A Machine Learning view of Distribution Estimation:

Efficient Computation of Empirical Proper Losses, Mixed in/out-of-sample Asymptotics of Empirical Proper Losses, a Unified Machine Learning Interface for Density Estimation, and a Systematic Benchmarking Experiment

Nurul Ain binti Toha

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Statistical Science
University College London

February 21, 2022

I, Nurul Ain binti Toha, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Probability distribution is a fundamental area in Statistics. It provides an understanding of the behaviour of a dataset. Distribution estimation is a task to estimate the distribution of a dataset. In machine learning, distribution estimation has been viewed as an unsupervised task as it uses unpaired datasets. One of the focuses of this thesis is to frame, explore and investigate distribution estimation as a supervised learning task (Chapter 3). The goal is to learn a function using an unpaired dataset to predict the distribution of the dataset. Loss functions are used to evaluate the accuracy of the prediction with respect to the true value. In the supervised distribution estimation task, a loss function depends on the type of estimator because it compares each input data points with its predicted distribution. Hence, we present an efficient method to derive the analytic expression of three probabilistic loss functions to evaluate the loss of standard kernel and kernel mixture distribution at an observation point (Chapter 5). The method uses the properties of kernel functions and elementary integration. Loss functions are also used for parameter tuning. We investigate the difference in the behaviour of in-sample and out-of-sample empirical loss functions: (1) log-loss; (2) probabilistic squared loss (PSL); using Gaussian kernel PDF estimator as the bandwidth goes to 0 and infinity (Chapter 6). To perform a consistent training, predicting and evaluation steps for distribution estimation in **R**, we investigate and implement a unified interface for distribution estimation and integrate it into the package **mlr3proba** (Chapter 7). Lastly, we conduct a benchmarking experiment to compare multiple distribution learners on multiple datasets and evaluate the learners using different log-loss, probabilistic squared loss (PSL) and integrated Brier loss (IBL) (Chapter 8). The best learner with the minimum out-of-sample empirical loss is selected and all the learners will be ranked using the results from evaluation.

Impact Statement

Distribution estimation is one of the most fundamental area in statistics. The results presented is this thesis are based on exploring distribution estimation in machine learning. The results of the thesis are hopefully useful for both inside and outside of the academic field. The impacts of this thesis are as follows.

1. (Unconditional) Distribution estimation is commonly categorized as an unsupervised learning task. Work presented in Chapter 3 describes how distribution estimation can be considered as a supervised learning task. In supervised distribution estimation, the objective is to train a function using a dataset to output a distribution. Building on that, the loss functions for distribution estimation evaluates the predicted distribution at a value. We show that by comparing the expected generalization loss of the predicted distribution with the expected generalization loss of the true distribution we are able to recover back Kullback-Leibler divergence and mean integrated squared error.
2. The work in Chapter 5 provides a efficient method to compute the probabilistic loss functions for kernel-based distribution at an observation point. The method is applicable for computing the loss for standard kernel distribution and kernel mixture distribution. For kernel mixture, the loss functions can expressed in terms of mixture component. From the method, a closed-form expression of the probabilistic loss of 11 symmetric kernel functions are derived and algorithms to use them evaluating the loss. The closed-form expressions are expected to give a more accurate results of the losses and less computation time.
3. The work presented in Chapter 6 provides a clear consequence of using in-sample and out-of-sample tuning methods for parameter selection of kernel-based distribution estimator using the log-loss and PSL. The result provides an understanding of how the in-sample tuning method may lead to a different optimal bandwidth from the out-of-sample tuning method for both simulated and real-world datasets. The selected bandwidth from the in-sample tuning method

lead to a higher empirical loss than the bandwidth from the out-of-sample tuning method when used on a new test data.

4. The work presented in Chapter 7 is expected to be provide two contributions. First, the derivations from Chapter 5 are included in **distr6** allowing users to obtain the L2-norm of PDF, L2-norm of CDF and L2-norm of CCDF of a kernel object distribution. Second, a unified machine learning interface for distribution estimation in **ml3proba** enable users to train, predict and evaluate distribution estimation. This provides a quick, easy to implement and consistent step for distribution estimation especially for new users.
5. The work presented in Chapter 8 shows the performance of different distribution learners on multiple real world datasets by using benchmarking experiment. The distribution learners are ranked from best to worst when applied to real world datasets.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Franz Kiraly and Yvo Pokern for their continuous support and advice during my PhD journey. During these years, they have been guiding me through my research journey and train me to be an independent researcher. Our weekly meetings to discuss research ideas have truly expanded my mind and understanding in this field and also in research.

Secondly, I would like to thank University Sains Islam Malaysia and Ministry of Higher Education Malaysia for funding me during my time in UCL. I am truly grateful for giving me this opportunity.

Additionally, I would like to express my gratitude to Raphael Sonabend for the many discussions on machine learning concepts, mathematics, research, and reviewing my thesis. I am very grateful that he gave me the opportunity to contribute the **distr6** and **mlr3proba** packages in **R** and thank you for the many discussions about the packages. Many thanks to Wan Mohamad Husni bin Wan Mokhtar for all the beneficial discussions on mathematical theory.

Thirdly, I am grateful to Xiaochen and Asma for their warm support and their help to review my writing. Our daily discussion and time together at the department will always be something I cherish. I would like to extend my gratitude to all of my friends, Nur Hayati, Nur Afifah, Liyana Adilla, Marco, Marcel, Yincheng, Iqbal, Rathish, Rasika, Suffeiya, Istvan and others. It is their kind help and support that have made my experience in UCL and London memorable.

Lastly, to my dearest parents and my wonderful siblings, I cannot express how thankful I am. Thank you for believing in me, for giving me the encouragement to move forward and for making me smile. Without their love, prayers, support and understanding, I would not be where I am now. I am forever in debt to them.

Contents

List of Figures

List of Tables

Abbreviations

Abbreviations	Meaning
t.v.i	Take value in
PDF	Probability density function
CDF	Cumulative distribution function
QF	Quantile function
ML	Machine learning
MSE	mean squared error
MISE	Mean integrated squared error
AMISE	Asymptotic mean integrated squared error
PSL	Probabilistic squared loss
IBL	Integrated Brier loss
LOOCV	Leave one out cross-validation

Table 1: Table of Abbreviations

Chapter 1

Introduction

Distribution estimation is useful to understand the nature of a dataset. This is not limited to just finding the mean, standard deviation or parameters. By understanding the distribution, we are able to know the pattern of the dataset including modality, symmetry, etc. Distribution estimation is the process of constructing a distribution from a dataset. A distribution is an object and is usually defined by the distribution defining functions: probability density function (or probability mass function) or cumulative distribution function. By knowing the distribution of the dataset, we are able to predict the probability an event happening.

Two of the important learning tasks in machine learning are: (1) supervised learning; (2) unsupervised learning. Supervised learning uses paired datasets (having features and label variables) and loss functions. In the supervised setting, the task is to learn a function that predicts the value of the target variable. Then, the loss function compares and evaluates the difference between the predicted and the true value of the target variable.

In machine learning, distribution estimation is commonly categorized as an unsupervised learning ([1], [2], [3]). This is because the learning task uses unpaired datasets and no loss functions for evaluation. In this thesis, we investigate distribution estimation as a supervised learning task where we learn a function that predicts the distribution of a dataset. We also investigate the probabilistic loss functions for distribution estimation. Different from the deterministic setting, the loss function in distribution estimation evaluates the defining function of the predicted distribution on the test points. Therefore, the computation of the loss functions requires the knowledge of the distribution. With the loss functions, we are able to use them for further investigation in distribution estimation including tuning and comparing different learners.

1.1 Objectives

Distribution estimation in machine learning is commonly considered as an unsupervised task due to the input data consists only of an unpaired dataset. To be considered as a supervised task, the aim of the task is to predict the value of the target variable of a paired dataset and there must exist a loss function to oversee the learning process. In classical statistics, Kullback-Leibler divergence and mean integrated squared error are used to measure the goodness of the estimated distribution and in parameter selection. [4] have discussed supervised learning for conditional distribution estimation. Motivated by [4], we explore and investigate unconditional distribution estimation from the perspective of supervised learning. Therefore, our first objective is to frame distribution estimation as a supervised learning task and we further investigate: (1) the relationship of mean integrated squared error of the estimated distribution with probabilistic squared loss and the integrated Brier loss; (2) the relationship of expected KL-divergence of the estimated distribution with the log-loss. Then, we discuss the link between probabilistic loss function for evaluating estimated distribution with MISE and expected KL-divergence from earlier literature. We also discussed some of the existing supervised learning algorithms used in distribution estimation.

The probabilistic loss functions for distribution estimation is a function of distribution defining function and an observation point. This is different compared to the supervised setting where a loss function measures the difference between a predicted value and a true value of the label variable. Hence, to evaluate the loss for distribution estimation task, the distribution defining functions are required. For kernel distribution, there are multiple kernel functions that can be used (e.g. Gaussian kernel distribution, Epanechnikov kernel distribution). Therefore, we want to investigate a method that can compute the probabilistic loss of a kernel-based distribution at an observation. Hence, the second objective of this thesis to provide an efficient method to compute the analytical expression of probabilistic loss functions: (1) log-loss; (2) probabilistic squared loss; (3) integrated Brier loss; to evaluate the loss for standard kernel distribution at a point and extending it to kernel mixture distribution.

In classical distribution estimation, Kullback-Leibler divergence and mean integrated squared error can be used to estimate the parameter (i.e. for non-parametric kernel methods, the parameter is the bandwidth). There are two important methods to select the parameter in distribution estimation. First, by minimising the asymptotic mean integrated squared error. Second, by minimising cross-validation estimate of Kullback-Leibler divergence and mean integrated squared error. The Kullback-Leibler divergence is the difference between the expected generalization log-loss of the estimated PDF and the true PDF. The mean integrated squared error is the difference between the expected generalization PSL of the estimated PDF and the true PDF. Therefore, log-loss and PSL can be used to estimate the Kullback-Leibler divergence and mean integrated squared error, respectively. Focussing on univariate kernel distribution, our third objective is to investigate the behaviour of in-sample and out-of-sample tuning methods using log-loss and PSL in distribution estimation.

Mathematical and statistical software allow a quick and easy way to implement methods to solve problems. There are many open source software that allow users to do statistical analysis. **R** is a useful statistical software that enable users to perform statistical analysis, modelling, solving equation and etc. In **R**, there are various packages that relate to statistical distribution. For example there is the **graphic::hist** function that enable user to estimate distribution via histogram. There is **stats::kde** that estimates point PDF using kernel methods. Then, there are modelling distribution packages that computes the PDF, CDF, quantile function (QF) and random numbers of a distribution but do not perform estimation. Then, we need to evaluate the estimated distribution using a loss function, for example using methods in the **scoringRule** packages. To perform machine learning train, predict and evaluate would need to use all of the functionality from different packages. However, these functionalities are not consistent in the syntax or command and produce different output. Due to these difference, our forth objective is to implement a unified machine learning interface for distribution estimation with the goal of providing users to perform machine learning task for distribution learners consistently.

Because distribution estimation is a well researched area, many learners for distribution estimation have been proposed over the years. This includes different estimators and different algorithm of parameter selection. Therefore, our fifth objective is to conduct a benchmarking experiment to compare and rank multiple distribution estimators on multiple dataset using different loss functions.

1.2 Contribution of thesis

The contribution of the thesis to achieve the objectives above is summarised below.

Framework for supervised distribution estimation

Chapter 3 frames distribution estimation as a supervised learning task. [4] discussed the framework on probabilistic supervised learning. In Chapter 3, we explain that for supervised distribution estimation, the task is to learn a function that predicts the distribution of unpaired datasets. We introduce probabilistic loss functions for distribution estimation that have already existed in literatures. These probabilistic loss functions are used to evaluate the estimated distribution at a point. In addition, we show that the divergence of the expected generalization loss function of estimated distribution and true distribution is equal to the expected Kullback-Leibler divergence and mean integrated squared error in classical distribution estimation. Therefore, to estimate the expected Kullback-Leibler divergence and mean integrated squared error only depends on the expected generalization loss of the estimated distribution since the expected generalization loss of the true distribution is unknown and can be taken as constant. Then, we discuss the use of probabilistic loss function in distribution estimation existed in literatures and relationship with the MISE and expected KL-divergence. Then, we briefly discuss the use of machine learning methods to estimate distributions.

An Efficient method to compute probabilistic loss functions

In Chapter 5, we proposed an efficient method to compute analytical expression of probabilistic loss functions (i.e. log-loss, probabilistic squared loss and integrated Brier loss) to evaluate the losses given a distribution and an observation point. In this thesis, we focus on kernel-based distribution. Using this method, closed-formed expression of the probabilistic loss functions of a kernel-based distribution at an observation point is obtained and can be used for evaluation. This method provides a general step to compute the analytical expression of any kernels and not focussing on just one. The method we used for this computation is by using the property of the kernel function and elementary integration. In this method, we show how to obtain all the terms to compute loss function for kernel function. The log-loss requires PDF and the probabilistic squared loss requires both PDF and L2-norm of PDF. The integrated Brier loss requires the L2-norm of CDF and L2-norm of the complementary CDF. We define the integration of kernel functions to obtain the kernel CDF and the partial L2-product of the kernel functions to compute the L2-norm of PDF.

Then using the kernel CDF, we find the L2-norm of the kernel CDF and CCDF by computing their partial L2-products. All of these functions can be extended to kernel mixtures. Once all the terms needed for loss functions are computed, we provide algorithms for computing the loss functions of homogeneous kernel mixture distribution. To complete this objective, we derived the analytical expression of the CDF, partial L2-products of kernel and partial L2-products of kernel CDF for most of the symmetric kernel functions which can be found in Appendix B.1.

Behaviour of in-sample and out-of-sample empirical probabilistic loss

In Chapter 6, we show the difference between in-sample tuning and out-of-sample tuning methods for parameter selection. There are two things we investigate: (1) the difference in the behaviour of in-sample and out-of-sample empirical log-loss in bandwidth tuning; (2) the behaviour of out-of-sample empirical PSL on different ratio of total test points in the test set to the observed data points in both training and test sets.

For the first investigation, we provide a formal proof using Gaussian kernel PDF to investigate the difference between the in-sample and out-of-sample empirical log-loss for bandwidth selection. We prove the limit of the out-of-sample empirical log-loss and in-sample empirical log-loss as the bandwidth goes to 0 and to ∞. The in-sample empirical log-loss tends to $-\infty$ as the bandwidth goes to 0 whereas the out-of-sample empirical log-loss tends to ∞ as the bandwidth goes to 0. Both in-sample and out-of-sample empirical log-loss will tend to ∞ as the bandwidth goes to ∞. It is found that only one new (unobserved) data point in the test set is needed for the out-of-sample empirical log-loss to be bounded (upper and lower) hence signifies that empirical out-of-sample log-loss have a minimum point. This minimum point reflects the optimal bandwidth.

For the second investigation, it is motivated by [5] and [6]. We provide a formal proof by giving a clear distinction in the use of training and test sets. We proved that for the out-of-sample empirical PSL of Gaussian kernel PDF to be bounded and achieved a global minimum, the ratio of total test points in the test set to the repeated (observed) data points in both training and test sets is $2\sqrt{2} : 1$. When exceed the ratio, the out-of-sample empirical PSL of Gaussian kernel PDF tends to ∞ and 0 as $h \to 0$ and $h \to \infty$, respectively.

To support the proofs, an experiment on 6 datasets is conducted to investigate the in-sample and out-of-sample tuning of bandwidth of a Gaussian kernel PDF via grid search using log-loss and PSL. Similar to result of the proofs, in-sample tuning results in choosing the smallest bandwidth when using log-loss and PSL. The out-of-sample empirical log-loss select the bandwidth with the minimum out-of-sample empirical loss for all datasets but this is not the case for all the datasets when using out-of-sample empirical PSL. This is due to the datasets have repeated data points. Further, we evaluated the tuned models using out-of-sample log-loss and found that the in-sample tuning methods for each dataset results to a higher loss compared to the out-of-sample tuning method.

Unified Machine Learning Interface for Distribution Estimation

In Chapter 7, we implement a unified machine learning interface for distribution estimation. The purpose for this is to allow a quick and easy implementation of distribution estimation using machine learning concepts. We use the platform that has already existed in **R**, which is **mlr3** package that provide unified interface for regression and classification. The related package, **mlr3proba** provides a unified interface for probabilistic setting using the same design interface as **mlr3**. We integrate the unified machine learning interface for distribution estimation in **mlr3proba**. We incorporate kernel PDF estimator and histogram PDF estimator into the **mlr3proba**. Other existing the distribution estimators in **R** can be found in **mlr3extralearners**. In total, there are 8 learners collected in **mlr3extralearners**. Then, we implement the loss functions as the score function to evaluate the distribution learners. The distribution learners can be use to train, predict and evaluate using the **mlr3** interface and also use all the **mlr3** extension functions such as tuning (in **mlr3tuning**), benchmarking and others.

Benchmarking Experiment for Distribution Estimation

In Chapter 8, we conduct a benchmarking experiment to investigate and compare the performance of different distribution learners with respect to the probabilistic loss functions. The objective is to compare and rank the distribution learners performed over all datasets. In this study, we use in total 54 datasets and 29 distribution learners. For all distribution learners, we evaluate their performance using log-loss. For all kernel based distribution learners, we further evaluate their performance on probabilistic squared loss. Additionally, for all Gaussian kernel based estimators, we evaluate their performance on integrated Brier loss. For evaluation, we re-sample the dataset via 3-fold cross-validation method. The average out-of-sample empirical generalization loss is computed. We analyse the results by: (1) averaging the loss over all datasets; (2) rank the learners based on the loss function for each dataset and average the rank over all datasets; (3) compare the learners using Friedman rank test. From the experiment, different probabilistic loss functions gave different results to the best ranked learner.

1.3 Thesis outline

The outline of the thesis is as follows. In Chapter 2, we provide a background on distribution as a mathematical object and the nonparametric estimators used to obtain distributions. In Chapter 3, we aim to frame distribution estimation as a supervised learning task, in which we provide a review of machine learning concepts in the first part of the chapter while the second half of the chapter will be on framing distribution estimation as supervised. Chapter 4 is a literature review focussing on estimating the Kullback-Leibler divergence, mean integrated squared error and bandwidth estimation for kernel distribution. Chapter 5 presents an efficient method to compute the probabilistic loss function for kernel based distribution to evaluate the loss given a kernel based distribution and an observation point. Chapter 6 is on the investigation of the behaviour of in-sample and out-of-sample empirical loss for bandwidth selection. Chapter 7 is based on unified distribution estimation in machine learning framework and Chapter 8 is on benchmarking experiments to compare nonparametric distribution estimation methods on empirical generalization losses. Finally, Chapter 9 is the conclusion and future work.

Chapter 2

Background: Distribution Estimation

2.1 Introduction

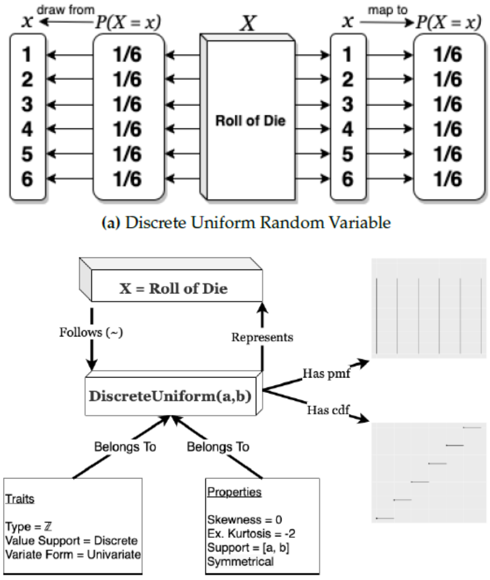
The main focus of this chapter is to provide a background review on the concept of probability distribution including its definition, types and estimators. This chapter is divided into two sections. In the first section, we define what we mean by 'distribution' and explain the types of distributions. In the second part, we discuss the estimator of a probability distribution focussing on nonparametric methods.

2.2 Distribution

In this section, we define probability distribution as mathematical objects and specify their properties and related operations. Then, we explain the different types of distributions for any random variable. Later, we define the functions that define a distribution (e.g. probability distribution function, cumulative distribution function). Throughout this thesis, we refer probability distribution as 'distribution' for short.

A distribution is a mathematical object on its own (a mathematical object is an abstract object). It has definitions, properties, traits and related mathematical operations. Distributions should be separated from random variables. [7] has recently explained distribution as a mathematical object and we will adopt that concept here. In Figure 2.1 [7], a clear distinction is made between distributions and random variables.

i. A random variable X (in Figure 2.1) has a distribution.
ii. A distribution represents a random variable.



(a) Discrete Uniform Random Variable

(b) Discrete Uniform Probability Distribution

Figure 2.1: Figure of a distribution as a mathematical object. In (a), the figure shows a random variable that follows a Uniform distribution. In (b), the figure shows how Uniform distribution representing a random variable. ([7]).

Therefore, as a mathematical object, the object distribution has the following.

i. A distribution is defined by functions, such as probability density function (PDF), cumulative distribution function (CDF), quantile function (QF) and etc.
ii. A distribution has properties such as mean, mode skewness and parameters.
iii. A distribution has characteristics to show whether it is discrete or continuous.
iv. A distribution has related operations.

Now consider a random variable X t.v.i $\mathbb{R}^n$ with a distribution d. Using [7]'s formulation, we specify the following.

i. Let distr$(\mathbb{R}^n)$ be a set of probability distribution for $\mathbb{R}^n$.
ii. Let d be the object distribution which is an element of distr$(\mathbb{R}^n)$.
iii. Let the CDF for d denoted as $d.F$ It is a function of (type) $d.F : \mathbb{R}^n \to [0, 1]$.
iv. Let the PDF of d denoted as $d.f$. It is a function of (type) $d.f : \mathbb{R}^n \to \mathbb{R}^+$.

2.2.1 Types of Distribution

Below, we briefly describe the different types of distribution of a random variable.

i. **Continuous:** A random variable X has a *continuous* distribution when X t.v.i $\mathbb{R}^n$ or in other words X has a continuous distribution if the probability density function (PDF) can be specified ([8]).

ii. **Discrete:** A random variable X has a *discrete* distribution when X t.v.i $\mathbb{N}$ or X has a discrete distribution if the probability mass function (PMF) can be specified ([8]).

iii. **Univariate:** A random variable X has a univariate distribution when X t.v.i $\mathbb{R}^n$ when $n = 1$ (for discrete X t.v.i $\mathbb{N}^n$ and $n = 1$).

iv. **Multivariate:** A random variable X has a multivariate distribution when X t.v.i $\mathbb{R}^n$ and $n > 1$ (for non-continuous X t.v.i $\mathbb{N}^n$ and $n > 1$).

v. **Parametric:** A random variable X is said to have a parametric distribution if we assume X follows a 'well-known' parametric family of distribution [5]. For example when X is sampled from a Binomial distribution with the parameter $N = 10$ (sample size) and $p = 0.6$ (the probability). In this example, X has a parametric distribution because Binomial is indeed a parametric distribution and the values of parameters are known.

vi. **Nonparametric:** A random variable X has a nonparametric distribution if it does not belong to any of the parametric family [5]. A distribution with the PDF plot in Figure 2.2 is an example where X has a nonparametric distribution.

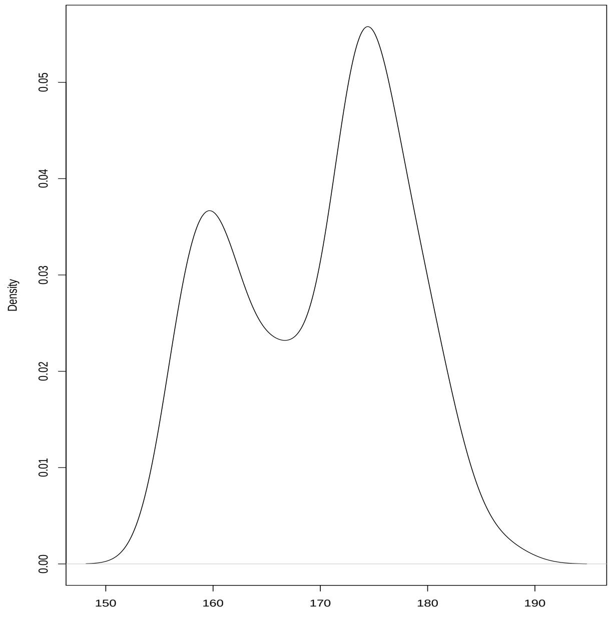


Figure 2.2: Figure of PDF plot for random variable of X which has a nonparametric distribution.

At first glance of Figure 2.2, it is difficult to characterize whether it is a parametric distribution or not. Therefore, we can consider it as *nonparametric* as a way to describe it as not belonging to any of parametric family of distribution.

vii. **Mixture:** Consider a vector of random variables, $X_1, \ldots, X_N$. Each X_i where $i = 1, \ldots, N$ is sampled from d_i distribution with the proportion w_i and $\sum_{i=1}^{N} w_i = 1$. Consider an example where $N = 2$. Let X_1 be the random variable that represents the height of female students with the distribution $N(160, 5)$ and X_2 is the random variable that represents the height of male students with distribution $N(175, 5)$. A random variable Y which is a random variable that represents the height of the students is the combination of X_1 and X_2 with the proportion 40% from the former and 60%, respectively. Then, we say that a random variable Y has a mixture distribution that has 40% chance following $N(160, 5)$ and 60% chance of following $N(175, 5)$.

2.2.2 Distribution Defining Functions

In this section, we properly introduce some functions that define a distribution. A distribution can be defined using multiple functions: (1) Probability density function (PDF); (2) Probability mass function (PMF); (3) Cumulative distribution function (CDF); (4) Moment generating functions (MGF) and others. However, not every distribution will have all of these functions. For continuous random variable, we define the PDF and CDF as in Def 2.2.1.

Definition 2.2.1. *Let X t.v.i $\mathbb{R}$ be a random variable. The probability density function (PDF) of X is*

$$p : \mathbb{R} \to \mathbb{R}^+ \tag{2.2.1}$$

where $\int_{-\infty}^{\infty} p(x)\, dx = 1$.
The cumulative distribution function (CDF) is

$$P : \mathbb{R} \to [0, 1] \tag{2.2.2}$$

where P is non-decreasing, $\lim_{x \to -\infty} P(x) = 0$ and $\lim_{x \to \infty} P(x) = 1$.

For discrete random variable, we define the PMF as in Def 2.2.2.

Definition 2.2.2. *Let X t.v.i $\mathbb{N}$ a discrete random variable. The probability mass function (PMF) of X is*

$$p : \mathbb{N} \to [0, 1] \tag{2.2.3}$$

where $p(x) > 0$ and $\sum p(x)\, dx = 1$.

It is important to note the following:

i. PDF exists only for continuous random variable. The CDF can be obtained by integrating the PDF. Therefore, when a distribution has PDF the CDF for the distribution exists.
ii. PMF exists only for discrete random variables. The CDF can be obtained by summation.
iii. A distribution can neither have a PDF or PMF but can have a CDF (e.g. Cantor distribution).
iv. For a categorical random variable, the CDF does not exist.

Mixture Distribution:
Consider univariate random variables $X_1, \ldots, X_N$. Recall again mixture distribution where each X_i where $i = 1, \ldots, N$ arises from different a distribution. Here, we provide a more formal definition for mixture distribution in Def 2.2.3.

Definition 2.2.3. *Let $x_1, \ldots, x_N \in \mathbb{R}$ be fixed observations. Let $P_i : \mathbb{R} \to [0, 1]$ and $p_i : \mathbb{R} \to \mathbb{R}^+$ be the CDF the PDF for each x_i is sampled from, respectively. A mixture CDF, G, and PDF, g, at x_i with weights $w_i > 0$ is*

$$g : x \to \sum_{i=1}^{N} w_i p_i(x - x_i) \tag{2.2.4}$$

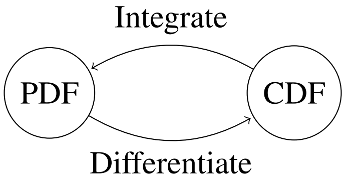
$$G : x \to \sum_{i=1}^{N} w_i P_i(x - x_i) \tag{2.2.5}$$

where $\sum_{i=1}^{N} w_i = 1$.. When $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$, then this is called Uniform Mixtures.

The mixtures in Definition 2.2.3 are finite. There are infinite mixtures but these will not be covered in here. A mixture distribution is known to be *homogeneous* if all the distributions p_i's (P_i's for CDF) are the same. When the distribution of p_i's (P_i's) are different, then it is known as *inhomogeneous* mixtures. A mixture distribution has a higher degree of freedom compared to a univariate uniform distribution, hence resulting a more flexible distribution.

2.3 Estimators for Distribution

In this section we explore and review some estimators of PDF and CDF for nonparametric continuous distribution. Suppose we have a dataset $\mathcal{D} = (X_1, \ldots, X_N) \overset{i.i.d}{\sim}$ where X t.v.i $\mathbb{R}^n$ and from an unknown distribution d with CDF $d.F$ and PDF $d.f$. n is the dimension of each X. For univariate random variable, $n = 1$. For simplicity, we write f for $d.f$ anf F for $d.F$. Given this random sample, we want to estimate the distribution, d. For a distribution of continuous random variable, the PDF and CDF can obtained by the relationship below.



i. A function f is of type $f : \mathbb{R}^n \to [\mathbb{R} \to \mathbb{R}^+]$.

ii. A function F is of type $F : \mathbb{R}^n \to [\mathbb{R} \to [0, 1]]$.

Histogram Estimator

A very well-known and easy to implement distribution estimator is the histogram. It is similar to a bar chart but the histogram is continuous. In the histogram algorithm, the collection of X_i, $i = 1, \ldots, N$, are first sorted in increasing order and then divided into groups, called bins. There are three important parameters in histogram estimator: (1) number of bins, B; (2) the width of the bins, w; (3) starting points of the histogram. These three parameters affect one another and control the shape of the distribution. B and w are related are related by Eqn (2.3.1,)

$$B = \frac{N}{w}. \tag{2.3.1}$$

where N is total number of data points. Consider using histogram PDF estimator. There are multiple ways to estimate PDF using histogram: (1) specifying the number of bins; (2) specifying the bin width; (3) start the histogram from 0; (4) start the histogram by the minimum X_i. We show the algorithm for estimating distribution via histogram PDF estimator as in Algorithm 1.

Algorithm 1 Estimating PDF by histogram

```
1: Inputs:
       Sample data: $\mathcal{D} = (X_1, ..., X_N)$
       Bins: $B_j$ where $j = 1, ..., M$
2: Outputs: Estimated PDF at the point $x$, $\hat{f}(x)$.
3: Steps:
4: Re-arrange $\mathcal{D}$ so that it is from minimum to maximum, i.e. $X_{(1)}, ..., X_{(N)}$ where
   $X_{(1)}$ is the minimum value in $\mathcal{D}$ while $X_{(N)}$ is the maximum.
5: Find the range, $R$, i.e. $R = X_{(N)} - X_{(1)}$
6: Divide $R$ by $M$ to find the width, $w_j$, of each bin such that, $w = \frac{R}{M}$
7: Count the number of $X_i$ in each $B_j$ and divide it by the $w$,
       $$\hat{g}_j = \frac{1}{Nw} \sum_{i=1}^{N} I(X_i \in B_j)$$
   where $I$ counts the number of $X_i$ inside each bin $B_j$ for $j = 1, \ldots, M$
8: Then, the estimated density centred at point $x$ by finding which bin $B_j$ for
   $j = 1, \ldots, M$ that $x$ belongs to and compute $\hat{f}(x) = \sum_{j=1}^{M} \hat{g}_j I(x \in B_j)$.
```

Naive Estimator

The naive PDF estimator is similar to histogram and based on counting the relative frequency of the observations in a small region,

$$\hat{f}(x) = \lim_{h \to 0} \frac{1}{2h} N_x \tag{2.3.2}$$

where N_x is the number of $X_i \in [x - h, x + h]$. Another representation of naive estimator is

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} w \left(\frac{x - X_i}{h} \right)$$

where

$$\delta(x) = \begin{cases} \frac{1}{2} & \text{if} \quad |x| < 1 \\ 0 & \text{otherwise.} \end{cases}$$

and δ is a delta function.

Empirical CDF

The empirical CDF (ECDF) is a step function that jumps up by $\frac{1}{N}$ at each point i for the random sample X. Formally, ECDF at a point x is defined as

$$\hat{F}_N(x) = P(X_i \leq x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\,(X_i \leq x) \tag{2.3.3}$$

where $\mathbb{1}$ is an indicator function such that

$$\mathbb{1}\,(X_i \leq x) = \begin{cases} 1 & \text{if} \quad X_i \leq x \\ 0 & \text{if} \quad X_i > x \end{cases}$$

that counts the number of X_i that are less than x. The algorithm for ECDF is shown in Algorithm 2.

Algorithm 2 Estimating CDF via ECDF

```
Inputs:
    Sample data: $\mathcal{D} = (X_1, ..., X_N)$
    A point $x$
Outputs: Estimated ECDF, $\hat{F}_N(x)$
Steps:
Re-arrange $\mathcal{D}$ from minimum to maximum, i.e. $X_{(1)}, ..., X_{(N)}$. where $X_{(1)}$ is
    the minimum value in $\mathcal{D}$ while $X_{(N)}$ is the maximum.
Count the number of $X_{(i)}$ that is less than $x$, $n = \sum_{i=1}^{N} \mathbb{1}(X_i \leq x)$
Compute the average over $N$, $\frac{n}{N}$
```

ECDF is similar to the histogram but without putting the observations into a bin (hence removing the parameters of the histogram i.e. bin width, number of bins) and each X_i is ordered in an increasing pattern (the value of $X_{(i)} < X_{(i+1)}$). A larger number of observation will lead ECDF to approach the true distribution for which $\mathcal{D}$ is sampled from. An example of ECDF plot is in Figure 2.3 where we can see the steps in between each point.

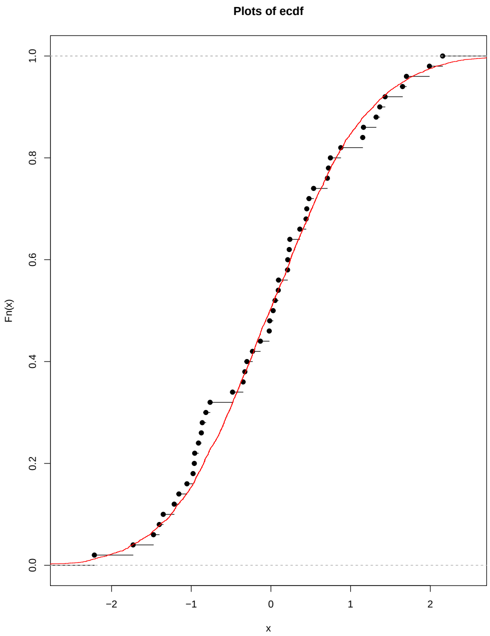


Figure 2.3: Plots of two different ECDF. The red line shows the plot of ecdf for $N = 5000$ and the black dash shows the plot for $N = 50$.

Kernel-based Estimators

Even though histogram is able to estimate PDF for continuous random variables, its bins are discrete and not smooth. One possible way to overcome this is to use kernel functions.

Just like histograms, the kernel based estimator is also a nonparametric method to estimate the PDF and CDF of a nonparametric distribution. The PDF and CDF follow the properties of the underlying kernel function (denoted as K). The kernel function K is a measuring tool to count the number of X_i of $\mathcal{D}$ that lies within the bandwidth h from the centre x. When K is a naive kernel function this will lead back to the naive estimator.

A kernel function not only acts as a weight function but also enforce continuity to the PDF and CDF. In addition, a kernel function on its own is also a PDF as it satisfies the following.

i. The area under the curve is 1: $\int K(u)\, du = 1$.
ii. Always positive: $K(u) > 0$.
iii. Under a certain region $\mathcal{A}$, the probability that any point $u \in \mathcal{A}$ is: $\int_{\mathbb{A}} K(u)\, du$.

i. is the most important must have property of a kernel function. This property ensures that the kernel can be used for probability distribution. Optional properties of kernel functions are as below:

i. Symmetric: $K(u) = K(-u)$

ii. Central mode: $K(u)$ reaches its maximum when $u = 0$.

The kernel PDF of a point x using the sample $X_1, \ldots, X_N$ is

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - X_i}{h}\right) \tag{2.3.4}$$

where $h \in \mathbb{R}$ is the bandwidth and N is the sample size. Here, h is the parameter that controls the degree of smoothness of the PDF and will need to be estimated (see Chapter 4 on methods to estimate the bandwidth). The kernel function K controls the weight of each X_i at the point x.

The kernel CDF is obtained by integrating its PDF, such that

$$\hat{F}(x) = \frac{1}{N} \sum_{i=1}^{N} I\left(\frac{x - X_i}{h}\right) \tag{2.3.5}$$

where

$$I(x) = \int_{-\infty}^{x} K(u)\, du \tag{2.3.6}$$

The choice of the bandwidth h plays a major role in determining the accuracy of the estimated PDF and CDF. However, the type of kernel function used does not play a significant role in the estimation ([9]). Table 2.1 shows frequently used kernel functions in kernel estimator.

Types of kernel	Kernel function
Uniform	$K(u) = \frac{1}{2}\mathbb{1}(\lvert u\rvert \leq 1)$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right)$
Epanechnikov	$K(u) = \frac{3}{4}(1 - u^2)\mathbb{1}(\lvert u\rvert \leq 1)$
Triangular	$K(u) = (1 - \lvert u\rvert)\mathbb{1}(\lvert u\rvert \leq 1)$

Table 2.1: Table of examples of kernel functions.

where $\mathbb{1}$ is an indicator function such that

$$\mathbb{1}(|u| \leq 1) = \begin{cases} 1 & \text{if} \quad u \leq 1 \\ 0 & \text{if} \quad u > 1. \end{cases}$$

K-Nearest Neighbor (KNN) Estimator

K-Nearest Neighbor (KNN) is another nonparametric method to estimate distribution. The algorithm to estimate the PDF at a point x using KNN is by ranking the distance between each sample data X_i and the point x.

Let $R_k^n(x)$ be the distance from x to the k-th nearest neighbour point X_i where n is the dimension of X, then the PDF at the point x is

$$\hat{f}_{KNN}(x) = \frac{k}{NV_n R_k^n(x)} \tag{2.3.7}$$

where $V_n = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$ and k is the parameter that controls the smoothness of the PDF. For univariate PDF estimation, the KNN estimator is

$$\hat{f}_{KNN}(x) = \frac{k}{2NR_k(x)} \tag{2.3.8}$$

[5] suggested to use $k = n^{1/2}$ while [10] suggested $k = N^{1/n}$, with n is the dimension of X_i while N is the size of the dataset.

Penalized Likelihood Estimator

Maximum likelihood estimation is normally used for estimating parameters. However, it was proposed by [11] and [12] and further explained in [13] to estimate the PDF using maximum likelihood estimation. This is done by estimating the entire curve. Consider a curve function $\hat{f}$. Initially, the log likelihood of $\hat{f}$ is

$$L(\hat{f}) = \sum_{i=1}^{N} \log \hat{f}(X_i). \tag{2.3.9}$$

Maximising the above equation leads to a rough solution of the mean of a set of Dirac functions at the N observations. To resolve this issue, subtracting the roughness of PDF $\hat{f}$ was proposed resulting maximising the likelihood function below,

$$L_\alpha(\hat{f}) = \sum_{i=1}^{N} \log \hat{f}(X_i) - \alpha R(\hat{f}'') \tag{2.3.10}$$

where $\alpha > 0$ and $R(\hat{f}'') = \int \left(\hat{f}''(x)\right)^2 \, dx$. α is a smoothing parameter that controls the roughness of $R(\hat{f}'')$ (i.e. small α causes $R(\hat{f}'')$ to be less smooth). The Eqn (2.3.10) has the advantage to control the goodness of fit of $\hat{f}$ (from the first term) and the smoothness (from the second term). Any function that maximises Eqn (2.3.10) is a *maximum penalized likelihood estimator*.

[14] introduced a variant of the penalizing factor. Rather than using $\int (\hat{f}'')^2 \, dx$, [14] suggested to use $R(\hat{f}) = \int \left(\frac{d}{dx}\right)^3 \log \hat{f}(x)^2 \, dx$. Using this approach, as $\alpha \to \infty$, the PDF estimator tends to a Normal distribution with mean and variance of the sample data. Furthermore, the log-likelihood function in Eqn (2.3.10) depends on the logarithm of $\hat{f}$, which tends to a positive estimated PDF. Then, [15] transformed $\hat{f}$ in Eqn (2.3.10) using logistic PDF. Distribution spline estimators are further developed by [16] and [17].

Other Distribution Estimators

So far, we discussed five family of nonparametric distribution estimators (i.e. histogram, kernel, KNN, ECDF, and penalized). Note that there are different variant of kernel methods including variable kernel ([5]). [18] proposed a log-spline approach where logarithm of PDF is modelled by spline. [19] proposed to use a wavelet approach in addition to explaining the criteria function for the wavelet PDF. [20] suggested to view mixture distribution estimator to be considered as another family of distribution estimator which consists of different base functions for each X_i. Another family of distribution is by using machine learning approach. [21] [22] proposed to estimate the PDF using neural network. [23] proposed to use decision tree to estimate PDF with similar methods by [24] and [25] but using hypercube.

Chapter 3

Distribution Estimation in Machine Learning

3.1 Introduction

The objective of this chapter is to provide a review on machine learning and to frame distribution estimation as a supervised learning task. Distribution estimation is a learning task that is commonly categorized as unsupervised learning ([1], [2]). In this section, we frame distribution estimation as probabilistic supervised learning.

This chapter consists of two parts. In the first part, we discuss the machine learning concepts:(1) supervised learning for regression and classification; (2) unsupervised learning; (3) meta-learning. This section is to give an introduction to the concept of machine learning and a clear distinction between supervised and unsupervised learning. The second part of this chapter is to frame distribution estimation as supervised learning. We first describe the setting where it can be viewed as supervised learning. We present the goal of supervised distribution estimation is to learn a function that predicts a distribution. Then, we show that the divergence between generalization loss of an estimated distribution and generalization loss of the true distribution is equal to the Kullback-Leibler and MISE in the distribution estimation setting.

3.2 Machine learning: Basic Concepts

This section is a review of machine learning concepts based on [3], [1] and [2]. In this section, we summarize and distinguish between supervised learning and unsupervised learning and also review meta-learning.

By looking at the types of learning task, machine learning can be categorized into supervised learning, unsupervised learning and reinforcement learning. We will focus the on the first two. In supervised learning, the goal is to learn a function f that maps the input variables to the output variable. The dataset in supervised learning consist of a paired dataset (i.e. for every input there is an output or a label). In addition, supervised learning has a loss function that enables a feedback mechanism which is use to evaluate if f maps the input correctly to output. In unsupervised learning, the task is to learn a function that is used to describe the relationships within the dataset. The datasets used consist of unpaired data.

This section is organized as follows. In Section 3.2.1, we describe supervised learning via regression, its aim, the properties of the estimators and how to evaluate the performance of the learning algorithms. This is followed by a discussion of classification task in Section 3.2.2. Then, we will discuss unsupervised learning in Section 3.2.3, including the aim and methods. Lastly, we review some meta-learning algorithms in Section 3.2.4.

3.2.1 Supervised Learning: Regression Tasks

The focus of this section is to provide a review on regression tasks. We define the setting, objective of the supervised learning and evaluation of the learning task.

3.2.1.1 Settings

For supervised regression tasks, we have the following setting. Consider a training dataset $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\} \overset{i.i.d}{\sim} (X, Y)$ where (X, Y) t.v.i $(\mathbb{R}^n \times \mathbb{R})$. Each X_i comprises the features (explanatory or independent) variables and Y_i is the target (response, label, dependent) variable. The goal of a supervised regression is to find a function $f : \mathbb{R}^n \to \mathbb{R}$ that is able to predict Y given X. A learning algorithm $\mathcal{A}$ is a process that uses the training dataset $\mathcal{D}$ to produce an estimator $\hat{f} = \mathcal{A}(\mathcal{D})$. $\hat{f}$ is an estimate of f. $\hat{f}$ is a good estimator if it can produce an accurate prediction of Y.

3.2.1.2 Function Estimation in Regression

In supervised learning, we assume there exists a relationship between X and Y. The relationship is modelled by an unknown function f,

$$Y = f(X) + \epsilon$$

where ϵ is the error term that is independent of X and has mean 0. However, most of the time only X is available and Y need to be predicted in which the prediction is based on the function f. Let $\hat{Y}$ be the predicted Y. The true f for any X and Y is unknown, hence there is a need to estimate it. Denoting $\hat{f}$ as the estimated function, then the relationship between $\hat{Y}$ and $\hat{f}$ is

$$\hat{Y} = \hat{f}(X).$$

However, it difficult to obtain $\hat{f}$ that is equal to f. A $\hat{f}$ is considered good if it is able to compute $\hat{Y}$ that is close to the real Y. Therefore, a measure is used to evaluate the accuracy $\hat{Y}$, by measuring the its difference with Y. We will discuss further in Section 3.2.1.3.

There are various estimators for f which we do not cover here because they are not the focus in this thesis. See [1], [2] and [3] for examples of estimators the for regression task.

3.2.1.3 Evaluating the Performance For Regression Tasks

In this section, we discuss evaluating the performance of the estimator $\hat{f}$ for a regression task. For a regression task and the training dataset $\mathcal{D}$, we can have multiple $\hat{f}$ to predict $\hat{Y}$ where different $\hat{f}$ will output different $\hat{Y}$. However, it is important to choose $\hat{f}$ that can best describe the relationship between X and Y of training dataset $\mathcal{D}$ to obtain the most accurate $\hat{Y}$ (i.e. the predicted value is close to the true value of Y). Evaluating the performance of $\hat{f}$ is by computing the discrepancy between Y and $\hat{Y}$ using loss functions.

Loss Function

A loss function is a tool that oversees the performance of the estimator $\hat{f}$. It is used to evaluate the performance of $\hat{f}$ by measuring the discrepancy between the true Y and the predicted $\hat{Y}$.

- In the regression setting, the loss function is a function of type $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
- It is important to evaluate the performance of $\hat{f}$ on an unseen dataset (test set). Let the test data be $(X^*, Y^*) \overset{i.i.d}{\sim} (X, Y)$. The generalization loss (test error, risk function or prediction error), is

$$R(\hat{f}) = \mathbb{E}_{X,Y} \left[\mathcal{L} \left(\hat{f}(X^*), Y^* \right) | \mathcal{D} \right]$$

The expectation is taken over (X^*, Y^*) and the training dataset here is fixed. This is known as the conditional generalization loss. The (conditional) generalization loss is only specific for a fix training data, $\mathcal{D}$.

- However, $\hat{f}$ is random and $R(\hat{f})$ is random due to the randomness of $\hat{f}$ (from to the randomness from $\mathcal{D}$).
- The *expected generalization loss* (also known as expected risk, unconditional generalization loss, expected test loss or expected prediction error) is,

$$\mathbb{E}_D[R(\hat{f})] = \mathbb{E}_{D,X,Y}\left[\mathcal{L}\left(\hat{f}(X^*), Y^*\right)\right] = \mathbb{E}\left[\mathcal{L}\left(\hat{f}(X^*), Y^*\right)\right]$$

Taking expectation on $R(\hat{f})$ remove the randomness of from $\hat{f}$. The expected generalization loss is the expectation of everything that is random. This is useful to evaluate the performance the learning algorithm $\mathcal{A}$ and considering the performance of the learning algorithm.

3.2.1.4 Examples of Loss Functions for Regression Task

For regression task, two loss functions that are normally used are shown below.

i. Squared loss,

$$\mathcal{L}_{sl}(y, \hat{y})) = (y - \hat{y})^2 \tag{3.2.1}$$

ii. Absolute loss,

$$\mathcal{L}_{abs}(y, \hat{y}) = |y - \hat{y}|. \tag{3.2.2}$$

The expected generalization loss of both can be obtained by taking expectation over the loss functions. The expected generalization squared loss is known as the mean squared error (MSE). The squared loss has the advantage to measure the bias-variance trade-off of a function f (see Section 3.2.1.5). The expected generalization absolute loss is the mean absolute error (MAE). The squared loss is better in detecting outliers compared to the absolute loss ([2]). In the presence of outliers in the predicted values, the difference between $\hat{y}$ and y will be large and the squared difference will be even larger. Therefore, the squared loss value is large in presence of outliers. In contrast, the absolute loss is not as good as the squared loss in detecting outliers [2] because the value of the absolute loss will not be as large as when using the squared loss.

3.2.1.5 Decomposition of the Generalization Loss

The expected generalization loss can be decomposed into bias and variance. To show that, we define the bias, variance and the trade-off between the bias and variance below for any estimator.

Definition 3.2.1. *Suppose θ is a parameter we want to estimate and its estimator is denoted as $\hat{\theta}$ is a random variable. Let the expectation of the estimator $\hat{\theta}$ be denoted as $\mathbb{E}[\hat{\theta}]$. Then, the bias and the variance for $\hat{\theta}$ are*

$$bias[\hat{\theta}] = \mathbb{E}[\hat{\theta}] - \theta \tag{3.2.3}$$

$$Var[\hat{\theta}] = \mathbb{E}\left[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2\right] \tag{3.2.4}$$

respectively.

Def 3.2.1 is a formal definition of bias and variance of a parameter. As in Def 3.2.1, *bias* is a measurement of discrepancy between the true unknown parameter and the expected estimated parameter whereas *variance* measures the expectation of squared difference between the estimated parameter and expected estimated parameter. An estimator is considered good when trade-off between the bias and variance are balanced. This is to avoid an estimator having an extremely high bias and an extremely low variance or vice versa. A good example to measure this trade-off is by using the mean squared error (MSE). We define the MSE for a parameter θ in Def 3.2.2.

Definition 3.2.2. *Mean squared error (MSE) of an estimator $\hat{\theta}$ for the parameter θ is the expectation of the squared distance between $\hat{\theta}$ and its true value*

$$MSE[\hat{\theta}] = \mathbb{E}[(\hat{\theta} - \theta)^2].$$

MSE can be decomposed into the bias and variance. For clarity, we show the decomposition in Proposition 3.2.1 below.

Proposition 3.2.1. *Let $\hat{\theta}$ be an estimator of θ, then the MSE of $\hat{\theta}$ is*

$$MSE[\hat{\theta}] = Var[\hat{\theta}] + bias[\hat{\theta}]^2$$

Proof.

$$\begin{aligned} MSE[\hat{\theta}] &= \mathbb{E}[(\hat{\theta} - \theta)^2] \\ &= \mathbb{E}\left[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta)^2\right] \\ &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] + (\mathbb{E}[\hat{\theta}] - \theta)^2 \\ &= Var[\hat{\theta}] + bias[\hat{\theta}]^2 \end{aligned}$$

where the bias and variance are defined in Def 3.2.1. □

Using the definitions of bias and variance together with Proposition 3.2.1, we show the decomposition of the generalization squared loss of the predictor $\hat{f}$ for the regression setting.

Proposition 3.2.2. *Let the training dataset be $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\} \overset{i.i.d}{\sim} (X, Y)$ where (X, Y) t.v.i $(\mathbb{R}^n, \mathbb{R})$. Let the test data be $(X^*, Y^*) \overset{i.i.d}{\sim} (X, Y)$. Let f be a function of type $f : \mathbb{R}^n \to \mathbb{R}$. Let $\mathcal{A}$ be a learning algorithm that produce $\hat{f} = \mathcal{A}(D)$. The expected generalization loss using the squared loss in Eqn (3.2.1) can be decomposed to*

$$\mathbb{E}\left[\left(\hat{f}(X^*) - Y^*\right)^2\right] = \mathit{Var}[\hat{f}(X^*)] + \mathit{Bias}^2[\hat{f}(X^*)] \tag{3.2.5}$$

Proof.

$$
\begin{aligned}
\mathbb{E}\left[\left(\hat{f}(X^*) - Y^*\right)^2\right] =& \mathbb{E}\left[\left(\left(\hat{f}(X^*) - \mathbb{E}[\hat{f}(X^*)]\right) + \left(\mathbb{E}[\hat{f}(X^*)] - Y^*\right)\right)^2\right] \\
=& \mathbb{E}\left[\left(\hat{f}(X^*) - \mathbb{E}[\hat{f}(X^*)]\right)^2\right] \\
& + 2\mathbb{E}\left[\left(\hat{f}(X^*) - \mathbb{E}[\hat{f}(X^*)]\right)\left(\mathbb{E}[\hat{f}(X^*)] - Y^*\right)\right] + \\
& \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(X^*)] - Y^*\right)^2\right] \\
=& \mathbb{E}\left[\left(\hat{f}(X^*) - \mathbb{E}[\hat{f}(X^*)]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(X^*)] - Y^*\right)^2\right] \\
=& \mathbb{E}\left[\left(\hat{f}(X^*) - \mathbb{E}[\hat{f}(X^*)]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(X^*) - \hat{Y}^*]\right)^2\right] + \\
& \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(X^*)] - \hat{Y}^*\right)^2\right] \\
=& \text{Var}[\hat{f}(X^*)] + \text{ Bias}^2[\hat{f}(X^*)] + \text{ Noise} \qquad (3.2.6)
\end{aligned}
$$

□

[1] refers to *bias* term as the "error that is introduced by approximating a real-life problem". It arises due to the model, $\hat{f}$, not being able to accurately fit the complicated real-world data. Meanwhile, the *variance* term is the "amount by which the estimator $\hat{f}$ would change if we estimate it using different training dataset". Therefore, a flexible estimator will have a higher variance with low bias compared to a less flexible method which will be the opposite. The noise term is known as the irreproducible error.

3.2.1.6 Properties of Loss Functions: Convexity

Consider a function g. The function g is *convex* such that when we take any two points x_1 and x_2 and evaluate them on g, the line that connects $g(x_1)$ and $g(x_2)$ lies above the graph function. A proper definition of a convex function is shown in Def 3.2.3 below:

Definition 3.2.3 ([26]). *Let g be a function where $g : \mathbb{R}^N \to \mathbb{R}$. Then, g is convex if the domain is a convex set and for all x_1 and x_2 in its domain, and $\lambda \in [0, 1]$ we* *have*

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda)g(x_2). \tag{3.2.7}$$

A continuous function is strictly convex if the equality of Eqn (3.2.7) holds with the additional condition $x_1 \neq x_2$ and $0 < \lambda < 1$. Convexity implies that any local minima is the global minimum of the function. Under strict convexity, the minimum point is unique.

Recall in the regression task, the goal is to construct a function f that best models the data $\mathcal{D}$. A loss function $\mathcal{L} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$ is strictly convex if $\mathcal{L}(\hat{f}(X), Y)$ is strictly convex for all $y \in \mathbb{R}$. If the loss function is strictly convex, then the generalization loss is also *strictly* convex (see [27]). The squared loss is an example of strictly convex loss.

3.2.2 Supervised Learning: Classification

In this section, we review another task in supervised learning which is the classification task. We define the setting, the function estimator, its properties and loss functions used to evaluate the performance of a classifier.

3.2.2.1 Setting

For classification task, we have the following setting. Consider the training dataset, $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\} \overset{i.i.d}{\sim} (X, Y)$ where X t.v.i $\mathbb{R}^n$ and Y is a type of discrete variable. For simplicity, we let Y t.v.i $\{0, 1\}$ (binary classification task). The goal of the supervised learning is to construct a function of type $f : \mathbb{R}^n \rightarrow \{0, 1\}$. Then, $\mathcal{A}$ is a learning algorithm that trains $\mathcal{D}$ to produce an estimator $\hat{f} = \mathcal{A}(\mathcal{D})$.

Classification task is not restricted to binary task. There are other types of classification methods including multi-class, multi-label and imbalanced classification. However, we do not discuss them here.

3.2.2.2 Function Estimation

Classification task is a type of supervised learning because of the existence of the target variable Y. The variable X and Y are related by the function f. The function f maps X to the class label. For the setting in Section 3.2.2.1, for each value of

X can either be mapped to 0 or 1 (binary case). Therefore, classification task is a method to group data.

Functions for classification task are called *classifiers*. A classifier can be categorized into two:

i. **Classical classifier:** This is the classical function that maps X to the class label. From Section 3.2.2.1, a label classifier is $f : \mathbb{R}^n \to \{0, 1\}$ where the target variable is binary.

ii. **Probabilistic classifier:** This is a more 'generalized' classifier. Instead of predicting the label of the target variable, it predicts the probability of the target variable given the feature variable. This uses conditional distribution function, $f : \mathbb{R}^n \to [\{0, 1\} \to [0, 1]]$ (for binary label). The return is a type of distribution defining function. For example, using the setting in Section 3.2.2.1, a classifier $\hat{f}$ predicts the probability for any given X_i that it belongs to $Y = 0$ and $Y = 1$.

3.2.2.3 Evaluating the Performance For Classification Task

This section is a discussion on how to evaluate the performance of a classifier. Similar to the regression task, a loss function is required to oversee the performance of the classifier. In this section, we discuss the loss functions for classification.

Loss Functions

Recall that classification is a mapping $f : \mathbb{R}^n \to \{0, 1\}$ or $f : \mathbb{R}^n \to [\{0, 1\} \to [0, 1]]$, i.e. mapping to the class label or to the probability of the class label, respectively. For each type of mapping, the loss function is shown in Table 3.1.

Type of classifier	Classifier	Loss function
Classical classifier	$f : \mathcal{X} \to \mathcal{Y}$	$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
Probabilistic classifier	$f : \mathcal{X} \to [\mathcal{Y} \to [0, 1]]$	$\mathcal{L} : [\mathcal{Y} \to [0, 1]] \times \mathcal{Y} \to \mathbb{R}$

Table 3.1: Table of functions for classifier loss functions. Here we use the notation where $X \in \mathcal{X} \subset \mathbb{R}^n$ and $Y \in \mathcal{Y} = \{1, \ldots, C\}$.

Consider the classical classifier in Table 3.1. The loss is incurred when the predicted class is not equal to the true class. That is, for binary case, the loss is 1 if $\hat{Y} \neq Y$.

For classical classifier, the loss measures the difference between the predicted label and the true label. For binary probabilistic loss, the loss function measures the difference between the true label and the probability of predicted label. An example of probabilistic loss is the Brier loss in Def 3.2.5. The expected generalization classification loss is computed by taking expectation of the loss function.

3.2.2.4 Examples of Loss Function for Classification

In this section, we provide some of the frequently used loss functions for the classification task. We group the loss function into: (1) loss function for label classifier; (2) loss function for probabilistic classifier.

Loss functions for the Classification Setting: Label Classifier

Examples of loss functions for classifiers are below.

i. 0-1 loss (for binary classifiers)

$$\mathcal{L}_m(y, \hat{y}) = I(y \neq \hat{y}) \tag{3.2.8}$$

ii. Hinge loss (for support machine vector)

$$\mathcal{L}_{hinge}(y, \hat{y}) = \text{max}(0, (1 - y)\hat{y}). \tag{3.2.9}$$

The 0-1 loss in Eqn 3.2.8, also known as the misclassification loss, is a standard loss function for classification tasks. As it suggested, a loss of 1 is incurred when the output of the learning function is not equal to the the true label. 0-1 loss is robust to outliers. However, this loss function is seldom used because it is non-continuous (not differentiable at 0) and non-convex (refer [28] for further discussion). The hinge loss in Eqn 3.2.9 is another classification loss normally for support vector machine (SVM) where it uses the max-margin property [2], [1]. For simplicity, we consider the binary SVM, where the classes of a label variable is separated by a hyperplane. There exist margins on the left and right side of the hyperplane. The hinge loss penalizes prediction based on two things, incorrect predictions and unconfident predictions (i.e. the loss can occur based on these two cases). The unconfident predictions are the predicted values that lie inside the margins. Therefore, the hinge loss is 0 only when the prediction is on the correct side whereas it is non-zero when prediction is on the wrong side or within the margin ([1]).

Loss functions for classification setting: Probabilistic classifier

So far, the examples of loss function shown are for predicting the class label. Throughout this section, we consider the probabilistic classifier where the output of f is a probability distribution of the class Y.

First is the logistic loss function in Def 3.2.4, also known as the cross-entropy loss. The logistic loss function is used to evaluate learners of logistic regression. The logistic function is similar to 0-1 loss but it compares the prediction distribution and the label. It can be seen a 'relaxation' of the 0-1 loss. It solves the continuity and convexity issue of the 0-1 loss.

Definition 3.2.4. *Let $f : \mathbb{R}^n \to P$ where $P \subseteq [\{0,1\} \to [0,1]]$ be a probabilistic logistic classifier. Then, the loss function is*

$$\mathcal{L}_{log}(P, y) = -\left(y \log P(y) + (1-y)\log(1-P(y))\right). \tag{3.2.10}$$

Another probabilistic loss function is the Brier loss, introduced by [29] to measure the accuracy of a weather forecast. This loss function is a squared distance between the predicted probability and the true class of Y (i.e. it is similar to the squared loss in the regression setting but with different inputs). The definition of Brier loss for binary variable is in Def 3.2.5.

Definition 3.2.5. *Let $\mathcal{P}$ be a set of CDF, such that $\mathcal{P} \subset [\{0,1\} \to [0,1]]$. Then, binary Brier loss, $\mathcal{L} : [P \times \{0,1\}] \to \mathbb{R}$, is defined as*

$$\mathcal{L}_{bl}(P, y) = \begin{cases} (1-P(y))^2 & \textit{if} \quad y = 1 \\ P(y)^2 & \textit{if} \quad y = 0 \end{cases}$$

which can be written as

$$\mathcal{L}_{bl}(P, y) = y(1-P(y))^2 + (1-y)P(y)^2. \tag{3.2.11}$$

The binary Brier loss in Def 3.2.5 is always between 0 and 1, where 0 means the probability is accurate while 1 means prediction is 100% untrue. A more general way of expressing the binary Brier loss by [30] is,

$$\mathcal{L}_{bl}(P, y) = (y - P(y))^2. \tag{3.2.12}$$

For probabilistic classification with multi-class problems ([29]), $X \in \mathbb{R}^n$ and $Y \in \{1, \ldots, C\}$ where $C \in \mathbb{N}$ where is C denotes the number of class. The classifier f is a function $f : \mathbb{R}^n \to \mathcal{P}$ where $\mathcal{P} \subseteq [\{1, \ldots, C\} \to [0, 1]]$. The multi-class Brier loss is

$$\mathcal{L}_{mbl}(P, y) = \sum_{j=1}^{C} (y_j - P_j(y))^2. \tag{3.2.13}$$

where $y_j = 1$ when the j^{th} event is observed and $y_j = 0$ otherwise.

An extension of Brier loss is the rank probability score ([31], [32],[33]), which is the squared distance of the predicted cumulative probability distribution and the cumulative observations.

3.2.2.5 Properties of Estimator Function in Classification

In this section, we review the bias and variance properties of a classifier. Then, we discuss how the bias and variance of a classifier relates to the bias-variance trade-off.

Recall the definition of bias in Def 3.2.1. Bias is the difference between expected prediction and its true value. However, in the classification task, when expected estimator is equal to the true estimator, the bias is 0 and vice versa. Therefore, the bias for classifier can only be two values, 0 or 1,

$$\text{bias}[\hat{f}(x)] = \begin{cases} 0 & \text{if} \quad \mathbb{E}[\hat{f}(x)] = f(x) \\ 1 & \text{otherwise} \end{cases}$$

The variance is a measure of expected difference between the prediction and the expected prediction value. It usually measures how much the model differs using a different dataset. The variance of a classifier is usually taken in the probability form, ([34])

$$\text{Var}[\hat{f}(x)] = P(\hat{f}(x) \neq \mathbb{E}[\hat{f}(x)])$$

The bias-variance decomposition was developed based on the squared loss for the regression task. The trade-off for classification is not as straight-forward as in regression. The bias-variance decomposition for classification task using the 0-1 loss functions is further discussed by [35], [36], [37], [38], [39], [34].

3.2.2.6 Properties of Probabilistic Loss Function for Classification: Proper

Properness ([40]) is a general property of evaluating *probabilistic* prediction (inlcuding probabilistic classification) and was first applied by [29]. As quoted in [41], properness is “to develop ways of motivating a forecast to be honest in the predictions he announces, and of assessing the performance of announced probabilities in the light of the outcomes then eventuate”. Properness of a generalization probabilistic loss function is defined in Def 3.2.6,

---

Definition 3.2.6. *Let $\mathcal{P}$ be a set of PDF, such that $\mathcal{P} \subset [\mathbb{R} \rightarrow \mathbb{R}^+]$ and let the loss function be $\mathcal{L} : \mathcal{P} \times \mathbb{R} \rightarrow \mathbb{R}$. Let Y t.v.i $\mathbb{R}$. Then the loss function is*

1. *proper if*

$$\mathbb{E}\left[\mathcal{L}(q, Y)\right] \leq \mathbb{E}\left[\mathcal{L}(p, Y)\right]$$

   for all $p, q \in \mathcal{P}$ and $Y \sim q$.

2. *strictly proper if $q, p \in \mathcal{P}$, the following are equivalent*
   (a) *$q = p$ and*
   (b) *$\mathbb{E}\left[\mathcal{L}(p, Y)\right] = \mathbb{E}\left[\mathcal{L}(q, Y)\right]$*

From Def 3.2.6, only the true distribution will minimize the generalization loss.

3.2.3 Unsupervised Learning

In this section, we review unsupervised learning by defining the setting and the aim of the learning and some methods that perform the unsupervised task.

Unsupervised learning consists of learning methods based on datasets that only have the feature variables. The objective of unsupervised learning is to explore the dataset by finding: (1) relationships between the data points; (2) relationships between variables; (3) more information and better understanding of the dataset. Thus, the objectives of unsupervised learning are different compared to supervised learning (which is prediction) but related. Unsupervised learning can be seen as a pre-processing stage of the dataset that is by understanding the properties of the dataset will help to make a better prediction.

Earlier, we discussed the use of loss functions which aid in evaluating methods by comparing the true and the predicted value of the target variable for supervised learning (or comparing the true value of the target variable and the predicted distribution of the target variable). In unsupervised learning, there is no loss function to oversee the performance of the estimator. However, unsupervised learning involves an optimizer that is used to estimate the parameters of the method.

3.2.3.1 Setting:

The setting for unsupervised is given here. Consider a training dataset $\mathcal{D} = (X_1, \ldots, X_N)$ where X t.v.i $\mathbb{R}^n$.

3.2.3.2 Objectives in Unsupervised Learning

This section is a summary review on some of the unsupervised learning objectives. The three main goals of unsupervised learning task are: (1) inference; (2) feature extraction; (3) density (distribution) estimation.

Inference

Inferencing is deducing population properties based on a dataset. This can include hypothesis testing and estimation. In supervised learning, inference occurs in the estimation of the parameters of a model. For unsupervised learning, inference is to explore the pattern or the distribution of the dataset. Clustering is an example of an unsupervised method that enables us to understand the structure of the dataset as it studies the relationship between points or between variables in the dataset.

Feature extraction

Consider dataset $\mathcal{D}$ in the setting in Section 3.2.3.1. Feature extraction is the process of reducing the dimension of $\mathcal{D}$ with n variables to another dataset $\tilde{\mathcal{D}}$ set with a lower number $\tilde{n} < n$ of variables, (i.e. the new dataset will have fewer columns than the original data set). In this method, all of the variables are used during the transformation process. The main purpose is to reduce the dimension of the original data set $\mathcal{D}$.

This method uses a function f that maps the current feature space to a new feature space, $f : \mathbb{R}^n \to \mathbb{R}^{\tilde{n}}$ where the new data set is denoted as $\tilde{\mathcal{D}} = (\tilde{X}_1, ..., \tilde{X}_N)$ and $\tilde{X}_i \in \mathbb{R}^{\tilde{n}}$. The process of extracting the features or transforming them is sometimes a crucial step before the features can be used in the supervised learning prediction stage. Feature extraction can also be supervised depending on the methods used. Examples of unsupervised learning methods are principal component analysis (PCA), factor analysis, clustering (for categorical data) and multidimensional scaling.

Distribution Estimation

Another goal of unsupervised learning is distribution estimation which is a subset of inference. Distribution estimation can also be viewed as supervised if it is conditional but for a non-conditional setting, it is considered as unsupervised. For the latter setting, where X t.v.i $\mathbb{R}$, is a univariate problem where X is from an unknown density function f. Then, in the unsupervised setting, the aim is to estimate the density of a point $x \in \mathbb{R}$, i.e. $\hat{f}(x)$. Methods of estimating density can be either parametric or nonparametric.

3.2.4 Meta - Learning

This section is a review on meta-learning and what it constitutes. Each meta-learning algorithm has its own purpose.

The concept of meta-learning is still vague and has no consistent meaning throughout literature. [42] defines meta-learning as a learning based on experience to understand the flexibility of the learning based on the domain and task. [43]'s view of meta-learning is “the understanding of the interaction between the mechanism of learning and the concrete contexts in which the mechanism is applicable”. [44] described meta-learning as a process that monitors the automatic learning process itself and tries to adapt its behaviour to perform better. [45] took in consideration from different literature including [42], [46], [43] and [44] and concluded that meta-learning is a learning system that includes base system that has experiences from learning on a single dataset or (and) different problems. In our perspective, meta-learning is used to improve learning process. In this section, we will focus on some of the meta-learning algorithms and discuss how they are considered meta-learning. Throughout this section, we will use the setting from the regression task as in Section 3.2.1.1 (or setting in Section 3.2.2.1 for classification task).

3.2.4.1 Meta-learning: Algorithm Recommendation or Parameter Tuning

Model selection is the process of selecting the best model by estimating the performance of multiple models ([2]). The performance of the models are evaluated by the empirical generalization loss. The best model will have the minimum empirical generalization loss. The process of tuning the parameter of a model is also a part of model selection. This method satisfied [45]'s definition of meta-learning where it learns from a single dataset for different problems. Different parameters input of a learning base model is considered as a different algorithm and will have different effects ([44]).

Meta-learning for tuning problem will learn from the input vector of parameters into the learning function and selects the optimal parameter with the minimum empirical loss. Tuning is performed during the training stage using the training set. However, from Section 3.2.1.3, this method will increase the chance of over-fitting since the training set is also being used for fitting. To avoid the problem of over-fitting, splitting the training set further where one set is for training and another is used from evaluation/tuning. This is known as out-of-sample tuning. The optimal parameter is chosen by minimising the empirical out-of-sample loss w.r.t the parameter. A more detailed discussion on tuning is in Chapter 6.

3.2.4.2 Meta-learning: Ensemble Learning

Meta-learning that learns from combination of algorithms is known as ensemble learning. The goal of ensemble learning is to combine multiple weak learners to produce a stronger one. In this section, we look into different ensemble learning algorithms.

Bagging

Bagging [47] also known as bootstrap aggregation is an ensemble learning algorithm which aims to reduce the variance of the learning method by repeated random sampling with replacement. The repeating re-sampling process and taking the average will reduce the variance. In bagging, the training set, $\mathcal{D}$, is re-sampled B times by bootstrapping (i.e. a random sampling method with replacement). Let $\mathcal{D}^b$ be a bootstrapped dataset for $b = 1, \dots, B$. A learning function, $\hat{f}$, is trained on each bootstrapped training set, $\mathcal{D}^b$, and each trained function $\hat{f}^b$, is then apply on the test set, $\mathcal{D}^*$ to produce B predictions, $\hat{f}^1, ..., \hat{f}^B$. The average of the predictions is then computed.

$$\bar{f}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x). \tag{3.2.14}$$

[46] and [44] constitute bagging as part of meta-learning because it uses knowledge from different learning tasks. The different learning tasks arise from repeated re-sampling with replacement which result to independent $\mathcal{D}^b$. The algorithm for bagging is shown in Algorithm 3.

Algorithm 3 Algorithm for Bagging

```
Inputs:
    Training data: $\mathcal{D} = ((X_1, Y_1), ..., (X_N, Y_N))$
    Test data: $\mathcal{D}^* = ((X_1^*, Y_1^*), ..., (X_M^*, Y_M^*))$
    Learning function, $\hat{f}$
Steps:
for $b = 1, ..., B$ do
    Resample $\mathcal{D}$ to obtain $\mathcal{D}^1, \dots, \mathcal{D}^B$ sets
    Train $\hat{f}$ on $\mathcal{D}^b$ to obtain $\hat{f}^1, \dots, \hat{f}^B$
    Use $\hat{f}^b$ for predictions on $\mathcal{D}^*$
end for
Output: Average prediction over $B$ as in Eqn (3.2.14).
```

Boosting

Boosting is another ensemble learning algorithm to improve prediction by combining weak learners sequentially to produce a stronger learner [2], [1], [48]. The concept used in this method is that a weak learner is better than just making a random guess [2]. By correcting and combining learners, the resulting new learner will be much better in predicting. Since the new learner depends on the previous one, the learners are not independent. There are many boosting algorithms that have been developed such as AdaBoost [49], gradient boosting [50] and stochastic gradient boosting [51].

Gradient boosting is related to gradient descent algorithm, such that for a function $h(\theta)$, gradient descent minimise h w.r.t θ by moving to the opposite direction of the gradient. Hence, the new θ_{i+1} is updated in each step by below:

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial h(\theta)}{\partial \theta_i}.$$

The general algorithm of gradient boosting has been discussed by [52], [53] and [51]. Gradient boosting by [50] can be broken down into three steps: (1) initialization; (2) projection of gradient learner; (3) line search ([53]). The algorithm for gradient boosting using the squard loss from Eqn (3.2.1) is shown in Algorithm 4.

Algorithm 4 Algorithm for gradient boosting for least square regression [50]

```
1: Inputs:
       Training data: $\mathcal{D} = ((X_1, Y_1), ..., (X_N, Y_N))$; Number of
       boosting steps: $B \in \mathbb{N}$; base learner: $h : \mathbb{R}^n \to \mathbb{R}$; Step size,
       $\alpha \in \mathbb{R}^+$
2: Output:$\hat{f}(X) = \hat{f}_b(X)$
3: Steps:
4: Initialize:
       Set $\hat{f}_0 = \frac{1}{N}\sum_{i=1}^{N} Y_i = \bar{Y}$
5: for $b = 1, \ldots, B$ do
6:     Compute the residuals for all $i = 1, \ldots, N$

           $\hat{e}_{ib} = Y_i - \hat{f}_{ib}(X_i)$

7:     Fit a base learner, $h_b$, using $X$ as the input and $\hat{e}_b$ as the input to output
   $\hat{h}_b(X)$
8:     Update $\hat{f}_b(X) = \hat{f}_{b-1}(X) + \alpha\hat{h}_b(X)$
9: end for
```

Algorithm 4 shows gradient boosting using squared loss in Eqn (3.2.1) and the step size α is constant and has been set. In the initialization stage, the first learner is set to be the mean of Y. The residuals are computed in step 6. In general, step 6 of Algorithm 4 which calculates the residuals is the negative derivative of a loss function $\mathcal{L}(Y_i, \hat{f}_b(X_i))$ w.r.t $\hat{f}_b$ at each X_i, i.e.

$$\hat{e}_{ib} = -\frac{d\mathcal{L}(Y_i, \hat{f}_b(X_i)}{d\hat{f}_b(X_i)}$$

for $b = 1, \ldots, B$ and $i = 1, \ldots, N$. Then, using the feature variable X and the residuals, $\hat{e}$ to fit a base learner and lastly the model $\hat{f}$ is updated as in step 8.

Stacking

Stacking [54] is a combination of base learners that are trained together on the same dataset. This method uses two levels of learner: (1) first level learners (base learners); (2) second level learners (meta-learners). In the fitting stage, a vector of first level learners $\hat{f}^k$ are trained using the training data, where $k = 1, \ldots, K$. The output of all the base learners will be a vector $(N \times K)$. This output will be considered as $\tilde{X}$ and combined with the original Y to be a new data set, $\tilde{\mathcal{D}} = (\tilde{X}_i, Y_i)$ where $i = 1, \ldots, N$ and $\tilde{X}$ and Y t.v.i $\mathbb{R}^K$ and $\mathbb{R}$, respectively. $\tilde{\mathcal{D}}$ will then be used to train the second level learners. Figure 3.1 shows the flow of stacking algorithm for training.

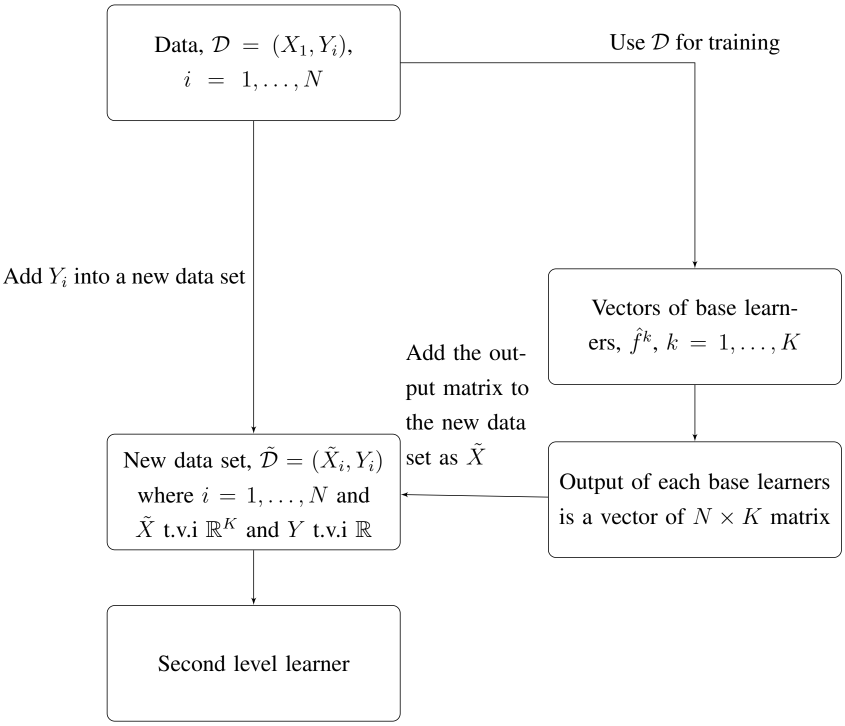

Data, $\mathcal{D} = (X_1, Y_i)$, $i = 1, \ldots, N$
Use $\mathcal{D}$ for training
Add Y_i into a new data set
Vectors of base learners, $\hat{f}^k$, $k = 1, \ldots, K$
Add the output matrix to the new data set as $\tilde{X}$
New data set, $\tilde{\mathcal{D}} = (\tilde{X}_i, Y_i)$ where $i = 1, \ldots, N$ and $\tilde{X}$ t.v.i $\mathbb{R}^K$ and Y t.v.i $\mathbb{R}$
Output of each base learners is a vector of $N \times K$ matrix
Second level learner

Figure 3.1: Figure for stacking algorithm.

3.3 Distribution Estimation as Supervised Learning Task

The aim of this section is to frame distribution estimation as supervised learning task. Distribution estimation has been categorized as an unsupervised learning task because the dataset used does not have a label variable. Hence, distribution estimation is mostly used for pattern recognition and inference. Consider a dataset that is sampled from an unknown distribution. In supervised distribution estimation, we aim to predict (estimate) the unknown distribution of sample.

Firstly, we define the setting where the task is estimating a distribution of an unlabelled dataset. Secondly, we define the bias and variance properties of the estimated distribution together with the bias-variance trade-off. The probabilistic loss functions evaluate the performance of the distribution at a point. Then, we derive the relationship between the expected generalization probabilistic loss functions and: (1) MISE; (2) KL-divergence; from the classical statistics.

3.3.1 Setting, Properties and Probabilistic Loss Functions of Distribution Estimation

In this section, we provide the setting for distribution estimation, the properties of a distribution estimator, probabilistic loss functions.

3.3.1.1 Setting

Let the training dataset $\mathcal{D} = (Y_1, \dots, Y_N) \overset{i.i.d}{\sim} Y$ where Y t.v.i $\mathbb{R}$ and Y follows the unknown distribution denoted by d. d is an element of distr($\mathbb{R}$) which is a set of distribution for $\mathbb{R}^n$. d can be defined using PDF and (or) CDF.

- The PDF of the distribution d is denoted as $d.f$, a function of type $d.f : \mathbb{R} \rightarrow \mathbb{R}^+$.
- The CDF of the distribution d is denoted as $d.F$, a function of type $d.F : \mathbb{R} \rightarrow [0, 1]$.

For simplicity, we use the notation of $f = d.f$ and $F = d.F$ for PDF and CDF, respectively. Unless stated otherwise, we assume that f and F represent the same distribution d.

The goal of the learning is to construct a function that maps $\mathbb{R}^N \rightarrow$ distr($\mathbb{R}$) so that $g(\mathcal{D})$ can predict a distribution $\hat{d}$, i.e. $g(\mathcal{D}) = \hat{d}$. To recover the PDF or CDF of the distribution by learning $g(D)$, we use the notation

$$g(\mathcal{D}) = \hat{f} \tag{3.3.1}$$

$$g(\mathcal{D}) = \hat{F} \tag{3.3.2}$$

We write $g(\mathcal{D}).f$ and $g(\mathcal{D}).F$ to denote that f and F obtained by learning from $g(\mathcal{D}$. g is an algorithm that learns using the training dataset $\mathcal{D}$ to output a predicted distribution $\hat{d}$, $g(\mathcal{D}) = \hat{d}$ where $\hat{d} \in$ distr$(\mathbb{R})$. Note that, since $\hat{f}$ and $\hat{F}$ are obtained from learning g on $\mathcal{D}$, both are also random. Unless stated otherwise, $\hat{f}$ and $\hat{F}$ are obtained by using the same algorithm g where $\hat{f}$ can be obtained by differentiating $\hat{F}$ while$\hat{F}$ can be obtained by integrating $\hat{f}$.

3.3.1.2 Properties of distribution estimators

For distribution estimation, the bias, variance and MSE follow similar structure as in Def 3.2.1 and Proposition 3.2.1. Therefore, we avoid repetition of formulating them here. However, bias, variance and MSE treat distribution estimation only measure the accuracy of the estimated PDF (or CDF) at a single point. This measurement is known as a local measure. Distribution estimation should be treated as a global problem which can be done by taking integration of the bias, variance and MSE w.r.t y to obtain integrated bias, integrated variance and mean integrated squared error (MISE). Unlike MSE, MISE can evaluate two things: (1) average global error; (2) accumulated point-wise error [55]. By integration, the conditional properties of the estimated distribution is reduced to unconditional properties.

The MISE for $\hat{f}$ is defined as,

$$\text{MISE}[\hat{f}] = \mathbb{E} \int \left(\hat{f}(y) - f(y)\right)^2 \, dy. \tag{3.3.3}$$

Since the integrand is non-negative, Fubini's theorem can be applied to MISE$[\hat{f}]$ and switch the position of expectation and integral to obtain

$$\text{MISE}[\hat{f}] = \int \mathbb{E}\left[\left(\hat{f}(y) - f(y)\right)^2\right] \, dy.$$

MISE can be decomposed into integrated squared bias (ISB) and integrated variance (IV) as shown in Proposition 3.3.1.

Proposition 3.3.1. *Let $\hat{f}(y)$ be an estimated PDF at a point y. Then, the MISE can be expressed as*

$$MISE(\hat{f}) = \int \left(\mathbb{E}[\hat{f}(y)] - f(y)\right)^2 \, dy + \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}\left[\hat{f}(y)\right]\right]^2 \, dy$$

Proof. The proof follows from Proposition 3.2.1. For clarity, we will show the steps as below.

$$\begin{aligned}
\text{MISE}[\hat{f}] &= \int \mathbb{E}\{\hat{f}(y) - f(y)\}^2 \, dy \\
&= \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}[\hat{f}(y)] + \mathbb{E}[\hat{f}(y)] + f(y)\right]^2 \, dy \\
&= \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}[\hat{f}(y)\right]^2 \, dy + \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}[\hat{f}(y)]\right]^2 \, dy \\
&= \int \left(\mathbb{E}[\hat{f}(y)] - f(y)\right)^2 dy + \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}[\hat{f}(y)]\right]^2 \, dy \qquad (3.3.4)
\end{aligned}$$

□

From Eqn (3.3.4) of Proposition 3.3.1, the first term is the integrated squared bias, ISB$[\hat{f}]$. It is the integral of the squared difference between the expected estimated PDF and the true PDF,

$$\begin{aligned}
\text{ISB}[\hat{f}] &= \int \left(\mathbb{E}[\hat{f}(y)] - f(y)\right)^2 \, dy \\
&= \int \text{bias}[\hat{f}(y)]^2 \, dy. \qquad (3.3.5)
\end{aligned}$$

The second term of Eqn (3.3.4) of Proposition 3.3.1 is the integrated variance, IV$[\hat{f}]$. It is the integral of the expected squared difference between estimated PDF and the expected estimated PDF,

$$\begin{aligned}
\text{IV}[\hat{f}] &= \int \mathbb{E}\left[\hat{f}(y) - \mathbb{E}\left[\hat{f}(y)\right]\right]^2 \, dy \\
&= \int \mathbb{E}\left[\hat{f}(y)^2 - 2\hat{f}(y)\mathbb{E}[\hat{f}(y)] + \mathbb{E}[\hat{f}(y)]^2\right] \, dy \\
&= \int \mathbb{E}[\hat{f}(y)^2] - \mathbb{E}[\hat{f}(y)]^2 \, dy \\
&= \int \text{Var}[\hat{f}(y)] \, dy. \qquad (3.3.6)
\end{aligned}$$

Hence, MISE can be expressed as

$$\text{MISE}[\hat{f}] = \text{ISB}[\hat{f}] + \text{IV}[\hat{f}]. \qquad (3.3.7)$$

The MISE is also applicable for measuring the bias variance trade-off for CDF estimator, $\hat{F}$,

$$\begin{aligned}\text{MISE}[\hat{F}] =& \int \mathbb{E}\left[\left(F(y) - \hat{F}(y)\right)^2\right]\, dy \\ =&\text{ISB}[\hat{F}] + \text{IV}[\hat{F}].\end{aligned} \tag{3.3.8}$$

In classical statistics, MISE is used to compare the true and estimated distribution ([55]). Another function is the Kullback-Leibler (KL) divergence,

$$D_{KL}(\hat{f}, f) = \int f(y) \log \frac{f(y)}{\hat{f}(y)}\, dy. \tag{3.3.9}$$

3.3.1.3 Probabilistic Loss Functions for Distribution Estimation

There are three main losses for distribution estimation: (1) log-loss; (2) probabilistic squared loss (PSL); (3) integrated Brier loss (IBL). These probabilistic loss functions have been introduced in literature. The log-loss was introduced by [56] ([41]) is defined in Def 3.3.1 below.

Definition 3.3.1. *Let $\mathcal{P}$ be a set of PDF , such that $\mathcal{P} \subseteq [\mathbb{R} \to \mathbb{R}^+]$. The log-loss $\mathcal{L} : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$ is defined as*

$$\mathcal{L}_{ll}(p, y) = -\log p(y). \tag{3.3.10}$$

In distribution estimation, the negative log-loss above (log-likelihood) has been used by [57] and [58] with leave-one-out cross validation to estimate the expected generalizaion loss [5] (see Chapter 4 on the use for bandwidth selection of a kernel PDF).

Probabilistic squared loss (PSL) (see [59], [60], [61]) defined in Def 3.3.2 is linked to the multi-class Brier loss. To support continuity, PSL uses integration instead of summation and PDF instead of CDF ([59]). [60] and [61] show that PSL is a strictly proper loss.

Definition 3.3.2. *Let $\mathcal{P}$ be a set of PDF , such that $\mathcal{P} \subseteq [\mathbb{R} \to \mathbb{R}^+]$. Then, probabilistic squared loss (PSL) $\mathcal{L} : [\mathcal{P} \times \mathbb{R}] \to \mathbb{R}$, is defined as*

$$\mathcal{L}_{psl}(p, y) = -2p(y) + \int p(y)^2 \, dy. \tag{3.3.11}$$

PSL is also known as Gneiting loss [60] and was used in distribution estimation by [62]. The generalization PSL is obtained by taking the expectation w.r.t Y for fix p as in Eqn (3.3.12).

$$\mathbb{E}[\mathcal{L}_{psl}(p, Y)] = \mathbb{E}\left[-2p(Y)\right] + ||p||_2^2. \tag{3.3.12}$$

A loss function derived from the rank probability score [31], [32], [33] and is also linked to Brier loss is the integrated Brier loss (IBL) or commonly known as the continuous rank probability score (CRPS) is defined in Def 3.3.3. In distribution estimation, the IBL was used by [63] in leave-one-out cross validion to estimate the expected generalization loss and subsequently in bandwidth selection of kernel PDF (see Chapter 4). This loss function is suitable for continuous and mixed random variables because it takes account the CDF rather than the PDF. The log-loss and PSL are only suitable for continuous random variables because both evaluates the loss of the estimated PDF at a point $y \in \mathbb{R}$. Mixed random variables have both continuous and discrete parts. The PDF of a mixed random variable will not always be defined (the PDF is not defined for the discrete part). Becuase both continuous and mixed random variables may be defined using the CDF, this makes IBL suitable to evaluate the distribution of both continuous and mixed random variables.

Definition 3.3.3. *Let $\mathcal{P}$ be a set of CDF, such that $\mathcal{P} \subseteq [\mathbb{R} \to [0, 1]]$. Then, integrated Brier Loss (IBL) $\mathcal{L} : [\mathcal{P} \times \mathbb{R}] \to \mathbb{R}$, is defined as*

$$\mathcal{L}_{ibl}(P, y) = \int \left(H(t - y) - P(t)\right)^2 dt \tag{3.3.13}$$

where $H(z)$ is the Heavy - side function,

$$H(z) = \begin{cases} 0 & \textit{if} \quad z < 0 \\ 1 & \textit{if} \quad z \geq 0. \end{cases}$$

Equivalently, we can re-write IBS as

$$\mathcal{L}_{ibl}(P, y) = \int_{-\infty}^{y} P(t)^2 \, dt + \int_{y}^{\infty} (1 - P(t))^2 \, dt. \tag{3.3.14}$$

3.3.1.4 Properness of the Probabilistic Loss Functions for Distribution Estimation

Probabilistic loss functions require to satisfy the properness property as in Def 3.2.6. All three probabilistic loss functions (log-loss, PSL and IBL) are strictly proper as stated in [60] and [61]. The properness of the log-loss and PSL are shown in Proposition 3.3.2.

Proposition 3.3.2. *Let $\mathcal{P}$ be a set of PDF such that $\mathcal{P} \subseteq \mathbb{R} \to \mathbb{R}^+$. Let $p, q \in \mathcal{P}$. Let Y be a random variable distributed by p.*

i. The log-loss in Def 3.3.1 is strictly proper.
ii. The PSL in Def 3.3.2 is strictly proper.

Proof. i. Recall the log-loss in Def 3.3.1. By the definition of properness from Def 3.2.6,

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{ll}(p, Y)] - \mathbb{E}[\mathcal{L}_{ll}(q, Y)] =& -\int \log(p(y))p(y) \, dy + \int \log(q(y))p(y) \, dy \\ =& \int p(y) \log \frac{q(y)}{p(y)} \, dy \\ =& D_{KL}(p, q) \end{aligned} \tag{3.3.15}$$

By Gibb's inequality, the divergence is KL-divergence $D_{KL}(p, q)$ is always more than or equal to 0 which leads to the properness. In addition to that, the log-loss is strictly proper if $q = p$.

ii. Recall the PSL in Def 3.3.2. Using Def 3.2.6,

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{psl}(p, Y)] - \mathbb{E}[\mathcal{L}_{psl}(q, Y)] =& ||q||_2^2 - 2\int q(y)p(y) \, dy + ||p||_2^2 \\ =& \int (p(y) - q(y))^2 \, dy \end{aligned} \tag{3.3.16}$$

The squared in the above divergence leads to non-negative, hence proper. When $q = p$, leads to a strictly proper loss.

□

Brier loss has been known to be a (strictly) proper loss. The IBL which is a derivation of the Brier loss is also (strictly) proper. Let $\mathcal{P}$ be a set of CDF such that $\mathcal{P} \subseteq \mathbb{R} \to [0, 1]$. Let $P, Q \in \mathcal{P}$. Let Y be a random variable distributed by P. Using Def 3.3.4, the divergence of the expected generalization IBL is $H_{L_{ibl}}(P, Q) = \mathbb{E} \int (P(t) - Q(t))^2 \, dt$. Since the integral is squared, the difference is positive and only 0 when $P = Q$.

3.3.1.5 Relationship between MISE & KL-divergence in Distribution Estimation and Probabilistic Loss Functions

In this section, we explain the relationship of MISE and KL-divergence for distribution estimation with the probabilistic loss functions in Section 3.3.1.3. In classical statistics, MISE in Eqn (3.3.3) and KL-divergence in Eqn (**??** are used to compare the predicted distributions with the true distribution and later used to select the best predicted distribution ([55]). There exist a relationship between the probabilistic loss functions in Section 3.3.1.3 and MISE and KL-divergence. The divergence of the expected generalization loss of the predicted distribution and expected generalization loss of the true distribution is able to recover MISE and KL-divergence.

Definition 3.3.4. *Let p and q be distribution functions. Let Y t.v.i $\mathbb{R}$ be a random variable distributed by p. Let $\mathcal{L}$ be a probabilistic loss function. The divergence of expected generalization loss of p and q using the loss function $\mathcal{L}$ is*

$$\mathcal{H}_{\mathcal{L}}(p, q) = \mathbb{E}[\mathcal{L}(p, Y)] - \mathbb{E}[\mathcal{L}(q, Y)]. \tag{3.3.17}$$

In Def 3.3.4, the loss function $\mathcal{L}$ evaluates the loss between p and random variable Y (and q with Y). p and q are fixed and the expectation is taken w.r.t the random variable Y (this is a total expectation). $\mathcal{H}_{\mathcal{L}}(p, q)$ is the divergence between the expected generalization loss function $\mathcal{L}$ of p and q.

The divergence $H_{\mathcal{L}}$ of the expected generalized loss function relates to the criteria functions via the following:

i. When $\mathcal{L}$ is log-loss as in Eqn (3.3.10), the divergence $\mathcal{H}_{\mathcal{L}}$ is equal to KL-divergence.

ii. When $\mathcal{L}$ is the PSL as in Eqn (3.3.11), the divergence $\mathcal{H}_{\mathcal{L}}$ is equal to MISE$[\hat{f}]$.

iii. When $\mathcal{L}$ is the IBL as in Eqn (3.3.13), the divergence $\mathcal{H}_{\mathcal{L}}$ is equal to MISE$[\hat{F}]$.

With right substitution, $H_{\mathcal{L}}(p, q)$ can be used to express the ISB and IV, shown in Proposition 3.3.3.

Divergence of Generalization Probabilistic Squared Loss and MISE of PDF

Recall the PSL as in Def 3.3.2.

Proposition 3.3.3. *Let $\mathcal{P}$ be a set of PDF such that $\mathcal{P} \subseteq [\mathbb{R} \to \mathbb{R}^+]$. Let $\mathcal{L}$ be a probabilistic loss function, $\mathcal{L} : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$. Let f be a PDF function, $f : \mathbb{R}^n \to [\mathbb{R} \to \mathbb{R}^+]$ and $\hat{f}$ is the estimate. Let the divergence of loss function $H_{\mathcal{L}}$ as in Def 3.3.4. Then,*

i. $\mathcal{H}_{\mathcal{L}_{psl}}(\mathbb{E}[\hat{f}], f) = \mathit{ISB}[\hat{f}]$

ii. $\mathbb{E}[\mathcal{H}_{\mathcal{L}_{psl}}(\hat{f}, \mathbb{E}[\hat{f}])] = \mathit{IV}[\hat{f}]$.

iii. $\mathcal{H}_{\mathcal{L}_{psl}}(\hat{f}, f) = \mathit{MISE}[\hat{f}]$

Proof. i. By substituting p and q with $\mathbb{E}[\hat{f}]$ and f, respectively, and PSL in Eqn (3.3.11) to $\mathcal{L}$ in Def 3.3.4 the divergence is

$$\begin{aligned}
\mathcal{H}_{\mathcal{L}_{psl}}(\mathbb{E}[\hat{f}], f) =& \mathbb{E}[\mathcal{L}_{psl}(\mathbb{E}[\hat{f}], Y)] - \mathbb{E}[\mathcal{L}_{psl}(f, Y)] \\
=& \left(\mathbb{E}[-2\mathbb{E}[\hat{f}]] + \int \mathbb{E}[\hat{f}]^2 \, dy\right) - \left(-2\mathbb{E}\left[f(Y)\right] + ||f||_2^2\right) \\
=& -2 \int f(y)\mathbb{E}[\hat{f}] \, dy + \int \mathbb{E}[\hat{f}]^2 \, dy + \int f(y)^2 \, dy \\
=& \int (f(y) - \mathbb{E}[\hat{f}])^2 \, dy \\
=& \text{ISB}[\hat{f}].
\end{aligned}$$

ii. In Def 3.3.4, the expectation is taken w.r.t the random variable Y. $\hat{f}$ is also random. By substituting p and q in Def 3.3.4 with $\hat{f}$ and $\mathbb{E}[\hat{f}]$, respectively and taking another expectation over $\mathcal{H}_{\mathcal{L}}(\mathbb{E}[\hat{f}], \hat{f})$,

$$\begin{aligned}\mathbb{E}[\mathcal{H}_{\mathcal{L}_{psl}}(\hat{f}, \mathbb{E}[\hat{f}])] =&\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)] - \mathbb{E}[\mathcal{L}_{psl}(\mathbb{E}[\hat{f}], Y)]\right]\\ =&\mathbb{E}\left[-2\mathbb{E}[\hat{f}] + ||\hat{f}||_2^2\right] - \mathbb{E}\left[-2\mathbb{E}[\hat{f}]\,dy + ||\mathbb{E}[\hat{f}]||_2^2\right]\\ =&\int \mathbb{E}[\hat{f}^2] - \mathbb{E}[\hat{f}]^2\,dy\\ =&\text{IV}[\hat{f}].\end{aligned}$$

The expectation taken over $H_{\mathcal{L}_{psl}}(\hat{f}, \mathbb{E}[\hat{f}])$ is to remove the randomness from $\hat{f}$.
iii. From Eqn (3.3.3), the MISE$[\hat{f}]$ can be obtained by adding ISB$[\hat{f}]$ and IV$[\hat{f}$. Then,

$$\begin{aligned}\mathbb{E}[\mathcal{H}_{\mathcal{L}_{psl}}(\hat{f}, f)] =&\mathcal{H}_{\mathcal{L}_{psl}}(\mathbb{E}[\hat{f}], f) + \mathbb{E}[\mathcal{H}_{\mathcal{L}_{psl}}(\hat{f}, \mathbb{E}[\hat{f}])]\\ =&\mathbb{E}[\mathcal{L}_{psl}(\mathbb{E}[\hat{f}], Y)] - \mathbb{E}[\mathcal{L}_{psl}(f, Y)] + \mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)] - \mathbb{E}[\mathcal{L}_{psl}(\mathbb{E}[\hat{f}], Y)]\right]\\ =&\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)]\right] - \mathbb{E}[\mathcal{L}_{psl}(f, Y)]\\ =&\mathbb{E}\left[-2\mathbb{E}[\hat{f}] + \int \mathbb{E}[\hat{f}(y)^2]\,dy\right] + ||f||_2^2\,dy\\ =&\mathbb{E}\int (\hat{f}(y) - f(y))^2\,dy \qquad (3.3.18)\\ =&\text{MISE}[\hat{f}] \qquad (3.3.19)\end{aligned}$$

□

There are a few things that we obtained from the result of Proposition 3.3.3.

i. Summing ISB$[\hat{f}]$ and IV$[\hat{f}]$, the term $\mathbb{E}[\mathcal{L}_{psl}(\mathbb{E}[\hat{f}], Y)]$ will cancel out and will obtained $H_{\mathcal{L}_{psl}}(\hat{f}, f)$.

ii. MISE$[\hat{f}]$ depends only the expected generalization PSL of the estimated PDF and expected generalization PSL of the true PDF.

iii. Re-arranging the result of MISE$[\hat{f}]$,

$$\mathbb{E}\left[\mathbb{E}[\mathcal{L}_{psl}(\hat{f}, Y)]\right] = \text{MISE}[\hat{f}] + \mathbb{E}[\mathcal{L}_{psl}(f, Y)]. \qquad (3.3.20)$$

shows that minimising the expected generalization PSL of $\hat{f}$, is equal to minimising the MISE$[\hat{f}]$ and the expected generalization PSL of the true PDF. However, due to the unknown expected generalization PSL of the true PDF, it can take it as a constant value. Therefore, when comparing two estimated PDF (e.g. $\hat{f}_1$ and $\hat{f}_2$), we only need to compare the expected generalization PSL of the estimated PDF.

iv. From Proposition 3.3.3, $\mathcal{H}_{\mathcal{L}}(\mathbb{E}[\hat{f}], f)$ is the bias term and $\mathbb{E}\left[\mathcal{H}_{\mathcal{L}}(\hat{f}, \mathbb{E}[\hat{f}])\right]$ $\mathcal{H}(p, q)$ is the variance component (we will show for IBL and log-loss below).

v. From Proposition 3.3.3, $\mathbb{E}[\mathcal{H}_{\mathcal{L}}(\hat{f}, f)]$ is decomposed to $\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}(\hat{f}, Y)]\right]$ and $\mathbb{E}\left[\mathcal{L}(\mathbb{E}[\hat{f}], Y)\right]$

Divergence of Generalization Integrated Brier Loss and MISE of CDF

Recall the IBL in Eqn (3.3.3). The expected generalization IBL for random variable Y is

$$\mathbb{E}[\mathcal{L}_{ibl}(F, Y)] = \mathbb{E} \int (F(t) - H(t-y))^2 \, dt.$$

Using the relationship of MISE of PDF with $H_{\mathcal{L}}$ in Proposition 3.3.3, $\mathcal{L}$ and PDF can be substituted with IBL in Eqn (3.3.13) and CDF, respectively, to show the following.

$$\begin{aligned}
\mathbb{E}[\mathcal{H}_{\mathcal{L}_{ibl}}(\hat{F}, F)] =& \mathcal{H}_{\mathcal{L}_{ibl}}(\mathbb{E}[\hat{F}], F) + \mathbb{E}[\mathcal{H}_{\mathcal{L}_{ibl}}(\hat{F}, \mathbb{E}[\hat{F}])] \\
=& \int \left(\mathbb{E}[\hat{F}] - F(t)\right)^2 \, dt + \int \mathbb{E}[\hat{F}^2] - \mathbb{E}[\hat{F}]^2 \, dt \\
=& \text{Bias}[\hat{F}] + \text{Var}[\hat{F}] \\
=& \text{MISE}[\hat{F}]. \qquad (3.3.21)
\end{aligned}$$

From the derivation above, $\mathcal{H}_{\mathcal{L}_{ibl}}(\mathbb{E}[\hat{F}], F) = \mathbb{E}\left[\mathcal{L}_{ibl}(\mathbb{E}[\hat{F}], Y)\right] - \mathbb{E}\left[\mathcal{L}_{ibl}(F, Y)\right]$ and $\mathbb{E}[\mathcal{H}_{\mathcal{L}_{ibl}}(\hat{F}, \mathbb{E}[\hat{F}])] = \mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{ibl}(\hat{F}, Y)] - \mathbb{E}\left[\mathcal{L}_{ibl}(\mathbb{E}[\hat{F}], Y)\right]\right]$.

Divergence of Generalization Log-loss and Kullback-Liebler Divergence

Recall the log-loss from Def (3.3.1). Its expected generalization loss is

$$\mathbb{E}[\mathcal{L}_{ll}(f, Y)] = -\mathbb{E}[\log f(Y)]. \qquad (3.3.22)$$

Using Proposition 3.3.3, Def 3.3.4 for log-loss can be expressed as

$$\begin{aligned}
\mathbb{E}\left[\mathcal{H}_{\mathcal{L}_{ll}}(\hat{f},f)\right] =&\mathcal{H}_{\mathcal{L}_{ll}}(\mathbb{E}[\hat{f}],f)+\mathbb{E}\left[\mathcal{H}_{\mathcal{L}_{ll}}(\hat{f},\mathbb{E}[\hat{f}])\right]\\
=&D_{KL}(\mathbb{E}[\hat{f}],f)+\mathbb{E}\left[D_{KL}(\hat{f},\mathbb{E}[\hat{f}])\right]\\
=&-\int f(y)\log(\mathbb{E}[\hat{f}(y)])dy+\left(\int f(y)\log(f(y))dy\right)\\
=&\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{ll}(\hat{f},Y)]\right]-\mathbb{E}\left[\mathcal{L}_{ll}(f,Y)\right]\\
=&\mathbb{E}\left[D_{KL}(\hat{f},f)\right]. \qquad (3.3.23)
\end{aligned}$$

From [64], $D_{KL}(\mathbb{E}[\hat{f}],f)$ is known as the bias component and $\mathbb{E}\left[D_{KL}(\hat{f},\mathbb{E}[\hat{f}])\right]$ is known as the variance component.

3.4 Discussion

Distribution estimation has always been viewed as unsupervised task in machine learning. This is because: (1) the dataset is unpaired; (2) loss functions are not properly discussed in the context of distribution estimation. However, distribution estimation has been linked to supervised learning in literatures such as using supervised learning algorithms to improve and estimate distribution estimation. [4] discussed supervised probabilistic prediction as a task focussing on conditional distribution. Viewing distribution estimation as a supervised learning task was motivated by [4] where univariate distribution estimation was proposed to be a subset of probabilistic supervised learning. In Section 3.3 of this thesis, we discussed the goal is to estimate a distribution of an unpaired dataset and the probabilistic loss functions that existed in literatures. We also derived the relationship between the expected generalization loss with MISE and expected KL-divergence.

In this section, we discuss and relate the use of probabilistic loss functions and distribution estimation. We discuss the use of loss function existed in distribution estimation and the relationships in Section 3.3.1.5 have been discussed in literature but was not clear. Then, we discuss some of the existing literatures that used supervised learning algorithm for distribution estimation.

3.4.1 Probabilistic loss function in distribution estimation

In this section, we discuss the use of probabilistic loss functions to evaluate estimated distribution has already existed in the literature.

In machine learning, loss functions are used to evaluate the goodness of the prediction. For example in the regression setting, the squared loss and the absolute loss measure the difference (error) between the predicted value with the true value of the target (lable) variable. In probabilistic prediction, score functions are used to evaluate probabilistic forecast ([60] and [41]). The score functions evaluates the performance of the distribution at a value it materialized ([60] and [41]). By definition, both loss functions and score functions are used to assess the performance of predictions. In Section 3.3.1.3 we use the term "probabilistic loss functions" which is the negative of the score functions (as in [60]). There is no difference in score functions and probabilistic loss functions, except the former are maximised whereas the latter are minimized. Therefore, we will continue to use the term"probabilistic loss function".

Recall that in machine learning, a "risk function" is the expectation of the loss functions with respect to the joint distribution of X and Y. Whereas, the "expected risk" or the "expected generalization loss" takes another expectation with respect to the training set. To estimate the generalization loss, cross-validation is used ([2]).

We want to highlight that in distribution estimation, the term loss function have been used but not in a consistent manner. The term loss function and risk functions have been used interchangeably because there is no clear definitions.

i. [62] stated that the log-loss and PSL are used to compute the loss from evaluating the PDF at a point
ii. [62] referred risk functions to MISE but also called $\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)]\right]$ and $\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{ll}(\hat{f}, Y)]\right]$ as risk functions
iii. [65] referred loss function to MISE and KL-divergence.

[65] viewed MISE and KL-divergence as loss functions because both compare the true PDF with the estimated PDF. However, this view has a disadvantage because the true PDF is usually unknown.

What is interesting is [62] stated that log-loss and PSL are used to evaluate the loss of from using the estimated PDF at a future point. This view is closely connected to the definition of loss functions and scoring rules. Therefore, the used of probabilistic loss function in distribution estimation have been proposed earlier to evaluate the estimated distribution. Since loss functions are used to oversee the learning, distribution can be categorised as a supervised learning task as it has probabilistic loss functions.

3.4.2 Using probabilistic loss functions to evaluate estimated distribution

Although [62] stated that log-loss and PSL may be used to compute the loss of the estimated PDF, many literatures in distribution estimation do not use the probabilistic loss functions for evaluation. In distribution estimation, the estimated PDF or CDF is compared with the true PDF or CDF, respectively, using MISE or KL-divergence. This is useful when datasets are simulated and we know the true distribution. Most literatures in distribution estimating used this method ([65], [66], [67], [68], [69], [70]).

However, when estimating the distribution of real world data, this method of evaluation may not be suitable since the true distribution in unknown. The use of proper probabilistic loss functions in Section 3.3.1.3 is more suitable. [60] explained that a probabilistic loss function is used to evaluate “the probabilistic forecast, by assigning numeric score based on the predictive distribution on the event or value it materialized”. Then how do we know the numerical value is good enough to tell whether our predicted distribution is good? The property of properness in Def 3.2.6 is important in this case. For example, given two PDFs, $\hat{f}$, $\hat{g}$ and a point $x \in \mathbb{R}$. The best PDF is the one with the minimum expected generalization loss (estimated using empirical loss). This support the use of log-likelihood for benchmarking experiment in distribution estimation. For example, [71] conducted a study of real-world data. The log-likelihood (negative log-loss) is used to compare different methods of PDF estimation. The PDF that has a higher log-likelihood is considered the best method (out of the methods being compared). SOme other literatures that use log-likelihood in experiments using real world data are [72] and [73].

Therefore, using the probabilistic loss functions for evaluation is appropriate for distribution estimation and comparing models. In later part of this thesis (Chapter 6 and Chapter 8), we use this evaluation method (using the probabilistic loss functions) for experimental investigation on real-world and simulated datasets.

3.4.3 Relationship between probabilistic loss function and KL-divergence and MISE

We have discussed the probabilistic loss functions that were proposed to evaluate the estimated distribution. In this section, we discuss the relationship between the expected generalization loss with MISE and expected KL-divergence. In Section 3.3.1.5, we showed the relationship between the divergence of expected generalization loss of the true distribution and the expected generalization loss of the estimated distribution is equal to expected KL-divergence and MISE (depending on the loss function used). This relationship is not new. In fact, this relationship was discussed earlier by [62] and [65] but was not obvious because:

- the loss function was not properly defined
- directly framed the relationship using cross-validation, i.e. to estimate the expected generalization loss.

Even though [62] and [65] discussed how to estimate MISE of PDF (and CDF) and KL-divergence, both have a different perspective but still arrived at in similar estimates.

Rudemo's method: We showed that MISE is the difference between expected generalization PSL of the estimated PDF and the expected generalization PSL of the true PDF. [62] proposed to estimate MISE by removing the term that contains unknown (true) PDF, to obtain

$$\int \mathbb{E}[\hat{f}(y)^2] - 2f(y)\mathbb{E}[\hat{f}(y)]\, dy \tag{3.4.1}$$

The above is the expected generalization PSL of the estimated PDF. And the term that is removed is actually expected generalization PSL of the true PDF. [62] also stated that taking the total expectation over the PSL and log-loss will give $\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)]\right]$ and $\mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{ll}(\hat{f}, Y)]\right]$, respectively.

Bowman's method:

The relationship in Section 3.3.1.5 appears in [65] and later in [63] but was not clear. This is because different loss functions were used. [65] referred the KL-divergence and integrated squared error (ISE), ISE$[p, q] = \int (p(y) - q(y))^2\, dy$, as loss functions, where p and q are PDF. This is a similar concept to the regression setting where the squared loss in Eqn (3.2.1) is the distance between the predicted value and the true value. However, [65] did not directly used the ISE but instead computed the difference between two ISEs, where

i. one ISE is the squared difference between dirac delta and the true and unknown PDF

ii. another ISE is the squared difference between dirac delta and the estimated PDF.

The difference of the two ISE is reduced to

$$\left(-2\hat{f}(y) + \int \hat{f}(y)^2\, dy\right) - \left(-2f(y) + \int f(y)^2\, dy\right) \tag{3.4.2}$$

If we take the total expectation of the above equation (over everything that is random), this is actually the MISE$[\hat{f}] = \mathbb{E}\left[\mathbb{E}_Y[\mathcal{L}_{psl}(\hat{f}, Y)]\right] - \mathbb{E}[\mathcal{L}_{psl}(f, Y)]$. However, [65] removed the unknown terms (the second bracket) and directly applied cross-validation to Eqn (3.4.2) which claimed to estimate MISE.

The derivation of Eqn (3.4.2) was also used for KL-divergence to compare the true and estimated PDF. [65] compared two KL-divergence, where

i. one is the KL-divergnce of dirac delta and the true and unknown PDF
ii. another is the KL-divergnce of dirac delta and the estimated PDF.

The difference between the two KL-divergence is $\log \frac{f(y)}{\hat{f}(y)} = \log f(y) - \log(\hat{f}(y))$. [65] directly applied leave-one-out cross-validation. If we take expectation over the randoms, we will have

$$\mathbb{E}\left[\mathbb{E}[-\log \hat{f}(Y)]\right] - \mathbb{E}[-\log f(Y)] = \mathbb{E}\left[\mathbb{E}[\mathcal{L}_{ll}(\hat{f}, Y)]\right] - \mathbb{E}\left[\mathcal{L}_{ll}(f, Y)\right] = \mathbb{E}\left[D_{KL}(\hat{f}, f)\right]$$

[63] later used this method to estimate MISE$[\hat{F}]$ using IBL as in Eqn (3.3.13) and taking the difference of $\mathbb{E}\left[\mathcal{L}_{ibl}(\hat{F}, Y)\right]$ and $\mathbb{E}\left[\mathcal{L}_{ibl}(F, Y)\right]$.

In this section, we discuss that the relationship in Section 3.3.1.5 actually exist in literature but we explain them in terms of expected generalization loss. However, the relationship was not that obvious because probabilistic loss function was not properly defined.

3.4.4 Distribution estimation and Supervised learning

In this section, we briefly discuss some literatures that attemps to link supervised learning and distribution estimation. Then, we discuss the use of ensemble learning for distribution estimation.

In machine learning, unconditional distribution estimation has been categorized as an unsupervised task. Some literatures attempt to view this as supervised learning. Supervised distribution estimation was proposed by [22] by estimating the PDF using neural network and framing the task as supervised by generating label data from a uniform distribution. Recently, [74] proposed a supervised learning PDF estimation by comparing the estimated distribution againts another distribution and using a modified log-loss as the loss function. Then, there are methods to estimate the PDF or CDF of a distribution using machine learning methods such as using decision tree ([23]), random forest ([75]) and neural network ([22], [76]), normalising flow as explained by ([77], [78]) and many more. These literatures focussed on methods of estimation. However, we want to focus more on the machine learning methods, specifically the ensemble learning - bagging, stacking and boosting. We discuss these methods because they are used to improve the prediction in the supervised learning by reducing the bias and variance whereas distribution estimation is considered as unsupervised. We focus on reviewing the ensemble learning because in distribution estimation, the IBS and IV are related to MISE and expected KL-divergence which is related to the probabilistic loss functions.

3.4.4.1 Ensemble learning Distribution Estimation

Ensemble learning methods are mainly used to improve training in supervised learning ([79]). The ensemble learning methods are useful to reduce the bias or variance. The use of ensemble learning to estimate PDF has been discussed by [80], [81], [48], [72], [52], [73], [79], [82], [71] and [83]. In this section, we discuss how the ensemble learning methods applied to distribution and effect of implementing ensemble learning to distribution estimation.

Bagging distribution estimation

[79] and [82] discussed the used of bagging to improve the estimated PDF of a distribution. [79] proposed three different types of aggregrating histogram algorithms: bagging of histogram, aggregrating histogram (using simple aggregration) and stacked histogram (motivated by [71]). The general algorithm below shows bagging for any PDF estimator by [82].

Algorithm 5 Algorithm for bagged distribution estimation ([82])

```
1: Inputs:
       Dataset, $\mathcal{D} = (Y_1, \dots, Y_N)$; PDF function,
       $\hat{f} : \mathbb{R} \to [\mathbb{R} \to \mathbb{R}^+]$; Number of bootstrap, $B$
2: Outputs: $\bar{f}(y)$.
3: Steps:
4: Resample $\mathcal{D}$ to obtain $\mathcal{D}^b$ for $b = 1, \dots, B$
5: for $b = 1, \dots, B$ do
6:     Train the PDF $\hat{f}$ using $\mathcal{D}^b$ to obtaib $\hat{f}^b$
7:     Estimate the PDF at $y$ using $\hat{f}^b$, $\hat{f}^b(y)$
8: end for
9: Average the PDF at $y$ over $B$, $\bar{f}(y) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(y)$
```

Algorithm 5 shows how to compute the bagged PDF estimation for any PDF estimator. The algorithm is not much difference compared to when bagging is applied to supervised setting (see [2]). To perform bagging, theh dataset is resampled B times. The PDF estimator is trained using each of the resampled data. Later, the trained PDF is used to estimate the PDF at new point.

Bagging is a learning algorithm that aims to reduce the variance. In regression setting, [2] showed that the MSE of bagged prediction is less than MSE of point prediction by reducing the variance.

[82] further investigated the effect of bagging on three different PDF estimators, i.e. the histogram, KDE and frequency polygon. [82] proved the L2-consistency for all three algorithms, i.e. $\mathbb{E}\left[(\hat{f}^b(x) - f(x))^2\right] \to 0$ as $h \to 0$, $N \to \infty$ and $Nh \to \infty$ where $\hat{f}^b(y)$ is the bagged estimated PDF at y. [82] also proved the L2-consistency of the bagged PDF.

Stacking distribution estimation

[71] and [83] used stacking to improve estimation of PDF. The method used for stacking PDF is by linearly combining multiple estimated PDF. The algorithm proposed by [71] for stacking PDF estimation and later used by [83] is shown below.

Algorithm 6 Algorithm for stacked distribution estimation ([71])

```
Inputs:
    Dataset, $\mathcal{D} = (Y_1, \dots, Y_N)$; PDF functions,
    $\hat{f}_k : \mathbb{R} \rightarrow [\mathbb{R} \rightarrow \mathbb{R}^+]$ where $k = 1, \dots, K$; Number of folds: $V$
Outputs: $\hat{f}(y)$.
Steps:
Split $\mathcal{D}$ into $V$ folds
for $v = 1, ..., V$ do
    for $k = 1, \dots, K$ do
        Train $\hat{f}_k$ on $\mathcal{D}^{-v}$ and estimates the PDF on $\mathcal{D}^v$
        Compute the linear combination of $\hat{f}_k(Y_i) = \sum_{k=1}^{K} \alpha_k \hat{f}_k(Y_i)$ for all $i = 1, \dots, N$
        Use EM-algorithm to find $\alpha_k$ for each $Y_i$, $i = 1, \dots, N$
    end for
end for
Update the combined PDF, $\hat{f}(y) = \sum_{k=1}^{K} \hat{\alpha}_k \hat{f}_k(y)$
```

In the stacking algorihtm above, PDF estimators are trained on the training dataset. These are the base learners. The PDF is estimated on each test fold. Since each fold will be used as test set, this will produce N estimated PDF which is evaluated using log-likelihood. For each Y_i, $i = 1, \dots, N$ there will be K PDFs and will be linearly combined. The combination of estimated PDF is the second level learner or meta-learner.

Stacking is proposed to reduce the bias ([54]). The stacking algorithm proposed by [71] did not commented on how staking KDE can reduce the bias of the estimated PDF. [83] (extend stacking KDE to multivariate distribution estimation) stated that stacking provides a better trade-off between bias and variance but no further investigation on how stacking reduces the bias and variance. This is something that needs more investigation. For distribution estimation, we want to reduce the IBS and see how this affects the trade-off between IBS and IV. [71] noted that when stacking individual kernel PDF, this result to a type of a mixture distribution.

Boosting distribution estimation

Boosting is an ensemble learning methods for supervised learning. It sequentially combines weak learners to produce a more powerful learner ([2]). In combining the weak learners, the boosting algorithm requires a loss function. [48], [80] and [52] proposed boosting strategies to improve the estimation of the PDF.

Gradient Boosting for distribution: [48] proposed a general boosting method for estimating the PDF of a distribution. This algorithm was then implemented by [73] for boosting histogram transform and [72] boosting Gaussian mixture. Let $\mathcal{D} = \{y_1, \ldots, y_N\}$. Let $f_1, \ldots, f_B$ be some base PDF estimators and $\mathcal{L}$ be some loss function. The idea for boosting algorithm is to built a PDF of $G(y) = \sum_{i=1}^{B} \alpha_b f_b(y)$ where G is the final PDF from boosting and each f_m is the weak PDF that is added in each boosting step. In the initialisation state, a weak base PDF estimator is set. What is important in boosting algorithm is how to add new weak learner. This is were the loss function plays the important role. [48] used the log-loss as the loss function. Generally, we want to add a base PDF estimator what will minimise $\sum_{i=1}^{N} \mathcal{L}(G, y_i)$. To do so in boosting step b, the gradient log-loss of $G(y) = G_{b-1}(y) + \alpha f_b(y)$ w.r.t G_{b-1} (using Taylor's expansion around G_{b-1}) to obtain

$$\sum_{i=1}^{N} \log(G(y_i)) \approx \sum_{i=1}^{N} -\log(G_{b-1}(y_i)) - \alpha \sum_{i=1}^{N} \frac{f(y_i)}{G_{b-1}(y_i)}.$$

Minimising $\sum_{i=1}^{N} \log(G(y_i))$ is the same as maximising $\sum_{i=1}^{N} \frac{f(y_i)}{G_{b-1}(y_i)}$. $\frac{1}{G_{b-1}(y_i)}$, for $i = 1, \ldots, N$ is the weight is that updated in each boosting step, b, i.e. $w_i = \frac{1}{G_{b-1}(y_i)}$. [48] noted that boosting algorithm is sequentially additive, hence, we cannot simply add new PDF at each boosting step (since by doing so we will have $G_b(y) = G_{b-1}(y) + a_b f_b(y)$ which may no longer be a distribution). In each boosting step b, the update PDF is $G_b(y) = (1 - \alpha_b) G_{b-1}(y) + \alpha_b f_b(y)$, $\alpha \in [0, 1]$. This boosting strategy is similar to [52] (but using Normal distribution).

In this boosting algorithm, it is not much different compared to the supervised gradient boosting. [48] proposed this general boosting method so that it can be applied to not just one PDF estimation.

Boosting was proposed to reduce the bias the predictions. However, [48] did not further investigate how the proposed boosting algorithm for distribution estimation relates to bias. From the experimental study, [48] only mentioned that boosting is more effective when combining weak learners.

Boosting KDE: [80] proposed another boosting method specifically for kernel distribution and [81] extend it to multivariate PDF estimation. The algorithm for boosting KDE is shown below.

Algorithm 7 Algorithm for boosting distribution by [80].

```
Inputs:
    Dataset, $\mathcal{D} : (Y_1, \ldots, Y_N)$; kernel function, $K_h : \mathbb{R} \to \mathbb{R}^+$;
    Number of boosting step: $B$; Normalising constant: $\alpha$
Outputs: $\hat{f}(y)$
Steps:
Initialize:
    Set $w_i^1 = \frac{1}{N}$ where $w_i^1$ is the weight for each $i = 1, \ldots, N$ for
    the 1st step
Set the bandwidth, $h$
for $b = 1, ..., B$ do
    Compute $\hat{f}_b(y) = \sum_{i=1}^{N} \frac{w_i^{(b)}}{h} K\left(\frac{y-Y_i}{h}\right)$
    Update the current weight for each $i$, $w_i^{(b+1)} = w_i^{(b)} + \log \frac{\hat{f}_{(b)}(Y_i)}{\hat{f}_{(b)}^{-i}(Y_i)}$
    Compute and normalized the boosted PDF, $\hat{f}(y) = \alpha \Pi_{b=1}^{B} \hat{f}_{(b)}(y)$
end for
```

Given a training dataset $\mathcal{D} = (Y_1, \ldots, Y_N) \overset{i.i.d}{\sim} Y$ and $Y \in \mathbb{R}$, the variable weight kernel PDF at $y \in \mathbb{R}$ is

$$\hat{f}(y) = \frac{1}{h} \sum_{i=1}^{N} w_i K\left(\frac{y - Y_i}{h}\right). \tag{3.4.3}$$

where w_i is the weight corresponding for each Y_i for $i = 1, \ldots, N$. Variable weight kernel PDF estimator signifies that each point has a different importance to the estimated distribution. Since boosting the is a combination of weak learners, the boosting algorithm is initialized by setting the initial weight is uniform over the dataset. As in [84], the bandwidth chosen and kept constant throughout the boosting steps.

The weight w_i is revised in each boosting step using $\log \frac{\hat{f}(Y_i)}{\hat{f}^{-i}(Y_i)}$ where $\hat{f}(Y_i)$ and $\hat{f}^{-i}(Y_i)$ are the PDF estimated at each Y_i while the latter is using the LOOCV. Therefore, the weight is updated by

$$w_i^{(b+1)} = w_i^{(b)} + \log \frac{\hat{f}_{(b)}(Y_i)}{\hat{f}_{(b)}^{-i}(Y_i)}$$

for $b = 1, \ldots, B$ steps and $i = 1, \ldots, N$. It is not clear why this log-likelihood ratio is used. However, by computation, [80] showed that at the second stage, the $w_i^{(2)} \propto \frac{1}{\hat{f}_{(1)}(Y_i)}$. This is the same weight that is obtained from [48].

The estimated PDF is multiplicatively combined by $\hat{f}(y) = \Pi_{b=1}^{B} \alpha \hat{f}_{(b-1)}(y)\hat{f}_b(y)$ where α is the normalizing constant to ensures $\hat{f}(y)$ integrates to 1. The method of combination is different from [48] that use linear combination ([79]).

This boosting algorithm for estimating kernel distribution proved that boosting reduced the bias at the second boosting step and reduction in bias is similar to [84]. However, to use kernel methods for boosting requires to 'weaken' the kernel PDF estimator is by increasing the bandwidth (oversmoothing). This is because kernel methods are flexible. Therefore, more consideration should be placed in choosing the bandwidth to make sure that it is not close to the true bandwidth. There are further investigations needed for this boosting algorithm: (1) study the effect of this boosting for steps greater 2 as suggested by the author; (2) comparing the boosting kernel methods with other bandwidth selection methods using real-world data. We can also investigate how the boosting method affects the variance of the estimated PDF. However, what is more interesting is whether the boosting algorithm reduces the IBS and how it affects the MISE.

Summary

In the section above, we discussed the use of ensemble learning methods for distribution estimation. This is not the first review on distribution ensemble learning. [79] did an overview and simulation experiments on the use of ensemble learning on distribution estimation (using simulated datasets and MISE for evaluation). Ensemble learning are strategies applied to supervised learning that can reduce the bias and variance. Using the ensemble learning are to improve the distribution estimation. However, there are more to investigate how these may benefits distribution estimation and how these relates to the integrated bias squared, integarted variance and MISE.

3.5 Conclusion

In this chapter, we frame univariate distribution estimation as a probabilistic supervised learning by explaining that the learning process is to find a function that predicts a distribution. The probabilistic loss function is used to evaluate the PDF or CDF of the estimated distribution. We derive the relationship of MISE and expected KL-divergence with the divergence of the expected generalization loss between the true and estimated distribution. We show that the ISB and IV can also be decomposed into the discrepancy between the expected generalization loss between the true and estimated distribution. PSL and IBL can be used to measure the bias-variance trade-off (for PDF and CDF, respectively). We also discuss the relationship of MISE and expected KL-divergence with the divergence of the expected generalization loss between the true and estimated distribution have existed in previous literature [65] and [63] but is not obvious. Further, we discussed the distribution estimation with supervised learning in literatures.

Chapter 4

Nonparametric Kernel Distribution Estimation

The objective of this chapter is to provide a review on the estimate of MISE and KL-divergence for a kernel distribution estimator. There are two important usage of MISE and KL-divergence in distribution estimation. The first is to compare and evaluate the goodness of different distribution estimator. Second is estimating the optimal parameter for distribution estimator by minimizing the estimated MISE or KL-divergence.

MISE for both PDF and CDF and KL-divergence of PDF are dependent on the true distribution which is unknown. Hence, several methods have been proposed to estimate the MISE and KL-divergence. In this chapter, we review on the estimation of MISE and KL-divergence via: (1) cross-validation method; (2) asymptotic method. In this chapter, we review by explaining and comparing the different methods in both categories for estimating of MISE and KL-divergence and further estimating bandwidth. Some of the methods are later compared in a benchmarking experiment in Chapter 8.

The outline of this chapter will be as follows. First, we provide the properties of the kernel estimator in Section 4.1 which are useful for estimating asymptotic MISE. Then, we review methods of estimating MISE for PDF and CDF in Section 4.2.

4.1 Properties of Kernel Distribution Estimator

In this section, we discuss the properties (i.e. bias, variance, MSE, IBS, IV and MISE) of the kernel distribution estimator for PDF and CDF. This section is a literature review from [5] and [55].

4.1.1 Properties for Kernel PDF

Let $Y_1, \ldots, Y_N \overset{i.i.d}{\sim} Y$ be a sample data and Y t.v.i $\mathbb{R}$ distributed from an unknown, d, distribution with PDF f and its respective CDF F which we are trying to estimate. Recall the kernel PDF from Eqn (2.3.4) with the kernel function K from Section 2.3. Before we proceed to define the properties of the kernel PDF, we assume that the properties of the kernel function K in Section 2.3 hold. In addition, K must also satisfy the following:

i. $\int uK(u)\, du = 0$
ii. $\int u^2K(u)\, du = \kappa_2 > 0.$

The properties of the kernel function K are important to show the bias, variance and MSE of kernels PDF. We use the usual notation, where f is the true PDF while $\hat{f}$ (with the hat) is the estimated PDF.

Let $\hat{f}$ be the estimated kernel PDF as in Eqn (2.3.4) and $\hat{f}$ follows the properties of the kernel function K. Then, the properties of $\hat{f}$ as in [14] are

i. Expectation of $\hat{f}$ is

$$\mathbb{E}[\hat{f}(y)] = f(y) + \frac{h^2\kappa_2}{2} f''(y) + \mathcal{O}(h^4). \tag{4.1.1}$$

ii. Bias of $\hat{f}$ is

$$\text{Bias}[\hat{f}(y)] = \frac{h^2\kappa_2}{2} f''(y) + \mathcal{O}(h^4) \tag{4.1.2}$$

where $f''(y)$ is the second derivative of f w.r.t y.

iii. Variance of $\hat{f}$ is

$$\text{Var}[\hat{f}(y)] = \frac{f(y)R(K)}{Nh} - \frac{R(f)}{N} + \mathcal{O}\left(\frac{h}{2}\right). \tag{4.1.3}$$

where $R(K) = \int K(u)^2\, du$ and $R(f) = \int f(y)^2\, dy$.

iv. The integrated square bias (ISB) of $\hat{f}$ is

$$\text{ISB}[\hat{f}] = \frac{h^4\kappa_2^2}{4} R(f'') + \mathcal{O}(h^6). \tag{4.1.4}$$

where $R(f'') = \int f''(y)\,dy$.

v. The integrated variance (IV$[\hat{f}]$), is

$$\text{IV}[\hat{f}] = \frac{R(K)}{Nh} - \frac{R(f)}{N} \tag{4.1.5}$$

where $R(K) = \int K(y)^2\,dy$.

vi. the MISE for $\hat{f}$ is the summation of Eqn (4.1.4) and Eqn (4.1.5)

$$\text{MISE}[\hat{f}] = \frac{h^2\sigma^2}{4} R(f'') + \frac{1}{Nh} R(K) + \mathcal{O}(h^6). \tag{4.1.6}$$

Detailed derivation of expectation and variance of $\hat{f}$ are in Appendix A.1 and Appendix A.2. The bias, variance and MSE of $\hat{f}$ are measurements at a single point y. The bias is low if the expected $\hat{f}(y)$ is close to its unknown true PDF. The bias can be reduced by increasing the variance and vice versa. Note that, i - vi depend on the unknown true PDF.

As we discussed in Section 3.3.1.2, distribution estimation should be treated as a global problem by taking integration w.r.t y for the bias, variance and MSE. The integrated square bias and integrated variance are shown in Eqn (4.1.4) and Eqn (4.1.5), respectively. The MISE of PDF is simply the integration on the squared of Eqn 4.1.2. By definition MISE in Eqn 3.3.3.

4.1.2 Properties of kernel CDF

Here, we define the properties of kernel CDF as reported by [85]. The derivations of the bias and variance of kernel CDF can be found in [86].

Let $\hat{F}$ be an estimated kernel CDF as in Eqn (2.3.5) and follows the properties of the kernel function K.

i. The expectation of $\hat{F}$ is

$$\mathbb{E}[\hat{F}(y)] = F(y) + \frac{h^2\kappa_2 F''(y)}{2} + \mathcal{O}(h^2) \tag{4.1.7}$$

ii. Bias of $\hat{F}$ is

$$\text{Bias}[\hat{F}(y)] = \frac{h^2\kappa_2 F''(y)}{2} + \mathcal{O}(h^2). \tag{4.1.8}$$

iii. Variance of $\hat{F}$,

$$\text{Var}[\hat{F}(y)] = \frac{1}{N}F(y)(1-F(y)) - \frac{\alpha}{N}f(y)h + \mathcal{O}\left(\frac{h}{N}\right) \tag{4.1.9}$$

where $\alpha = 2\int vI(v)K(v)\, dv$ and $F(y)$ is twice differentiable (see full derivation in [86]).

iv. MISE$[\hat{F}]$ is by taking integration of the MSE$[\hat{F}(y)]$ w.r.t y leading to

$$\begin{aligned}\text{MISE}[\hat{F}] =& \int \frac{h^4\kappa_2^2(F''(y))^2}{4} + \frac{1}{N}F(y)(1-F(y)) - \frac{\alpha}{N}f(y)h + \mathcal{O}\left(h^4 + \frac{h}{N}\right)\, dy \\ =& \int \frac{1}{N}F(y)(1-F(y))\, dx - 2\frac{h}{N}\int yK(y)I(y)\, dy + \\ & \frac{h^4}{4}\int x^2K(y)\, dy \int (F''(y))^2\, dy \end{aligned} \tag{4.1.10}$$

(as in [86] and [70])

Similar to the estimated PDF, i - vi above depend on the true unknown kernel CDF.

4.2 Estimation of MISE and KL-divergence for Bandwidth Selection

In this section, we review some methods to select bandwidth for kernel distribution. Bandwidth is important in kernel distribution as it can be one way to define the kernel distribution and it also controls the shape and smoothness of the PDF and CDF of the kernel distribution. Multiple methods have been proposed to estimate the bandwidth. In this section, we review some of the different methods that estimate MISE and KL-divergence which are later being used to estimate the bandwidth.

4.2.1 Estimation of MISE and KL-divergence via Cross-validation

In this section is a review on the estimation of MISE and KL-divergence using cross-validation. The loss functions (log-loss, PSL, IBL) has been used in distribution estimation by [62], [65], [63], [57], [64] empirically. For this section, we define datasets

$$\mathcal{D} = (Y_1, \ldots, Y_N) \overset{i.i.d}{\sim} Y \tag{4.2.1}$$

$$\mathcal{D}_{-i} = (Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_N) \overset{i.i.d}{\sim} Y \tag{4.2.2}$$

and Y t.v.i $\mathbb{R}$ where Y_i is not included in $\mathcal{D}_{-i}$. Let $\hat{f}_{-i}$ is the PDF estimator that estimates at the point Y_i on $\mathcal{D}_{-i}$.

The algorithm for estimating the MISE and KL-divergence via leave-one-out cross-validation (LOOCV) is shown below in Algorithm 8.

Algorithm 8 Algorithm for estimating MISE and KL-divergence via LOOCV

```
Inputs:
    Dataset, $\mathcal{D} : (Y_1, \ldots, Y_N)$;
    A kernel PDF function, $f : \mathbb{R} \to [\mathbb{R} \to \mathbb{R}^+]$;
    A loss function, $\mathcal{L} : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$.
Outputs: A numerical value for $\mathbb{E}_{emp}[\mathcal{L}(f, Y)]$.
Steps:
for $i = 1, ..., N$ do
    $\mathcal{D}_{-i} = (Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_N)$
    for $j = 1, \ldots, N-1$ do
        Fit the PDF at each $Y_i$ $f(Y_i) = \frac{1}{(N-1)h} \sum_{j=1}^{N-1} K\left(\frac{Y_i - Y_{j \neq i}}{h}\right)$
    end for
    Compute the empirical generalization loss, $\mathbb{E}_{emp}[\mathcal{L}(f, Y)] = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_{-i}, Y_i)$
end for
return $\mathbb{E}_{emp}[\mathcal{L}(f, Y)]$.
```

Likelikhood-cross validation: [57] and [58] proposed to use log-likelihood (or negative log-loss) by substituting Eqn (3.3.10) into $\mathcal{L}$ in the Algorithm 8. This leads to a negative LOOCV empirical log-loss, i.e.

$$\mathbb{E}_{emp}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)] = \frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{1}{h(N-1)} \sum_{j \neq i}^{N-1} K\left(\frac{Y_i - Y_j}{h}\right)\right). \tag{4.2.3}$$

The PDF is estimated at each Y_i using the remaining $N-1$ data points of $\mathcal{D}_{-i}$. This yields N PDF and evaluated by taking the average of N log-likelihood.

To estimate the bandwidth using this methods is by maximimising $\mathbb{E}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)]_{emp}$ as below.

$$h_{LCV} = \max_h \mathbb{E}_{emp}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)]. \tag{4.2.4}$$

By maximising $\mathbb{E}_{emp}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)]$, the predicted density is close to its true value.

The bandwidth selected by this method is related to Kullback-Leibler (KL-divergence) in which $\mathbb{E}_{emp}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)]$ minimises the KL-divergence between $\hat{f}$ and f. By taking negative $\mathbb{E}_{emp}[\mathcal{L}_{ll}(\hat{f}_{-i}, Y)]$ will provide an unbiased estimator for KL-divergence. [58] pointed out that if the point at which the density is estimated is equal to one of the sample data, the limit of the likelihood goes to $-\infty$ as the bandwidth tends to 0. Hence, LOOCV guarantees a non-zero PDF (this will be further discussed in Chapter 6 with a formal proof). This method has the advantage in which it is sensitive to outliers and is powerful to estimate PDF for a smaller sample size. However, it still is not suitable for long-tail data. [87] also claimed that this method is not suitable for data with long heavy property as it can cause inconsistent estimates.

LOOCV MISE PDF: The use of PSL for estimating the MISE$[\hat{f}]$ was proposed by [62] and [65]. Algorithm 8 produced an empirical generalization PSL which is similar to [65] as in Eqn (4.2.5).

$$\mathbb{E}_{emp}[\mathcal{L}_{psl}(\hat{f}_{-i}, Y)] = \frac{1}{N}\sum_{i=1}^{N}\int \hat{f}_{-i}(y)^2 dy - 2\hat{f}_{-i}(Y_i)\ dy. \tag{4.2.5}$$

[62] proposed an estimator of MISE by using in-sample data as in Eqn (4.2.6),

$$\mathbb{E}_{emp}[\mathcal{L}_{psl}(\hat{f}, Y)] = \int \hat{f}(y)^2 dy - \frac{2}{N(N-1)}\sum_{i\neq j}^{N} K\left(\frac{Y_j - Y_i}{h}\right). \tag{4.2.6}$$

[88] showed that $\int \hat{f}_{-i}(y)^2 dy = \int \hat{f}(y)^2 dy + \mathcal{O}\left(\frac{1}{N^2 h}\right)$. Eqn (4.2.6) is a more simpler to compute and does not involves any asymptotic effect ([68], [64]). The bandwidth estimated via this approach is by minimising $\mathbb{E}_{emp}[\mathcal{L}_{psl}(\hat{f}, Y)]$ of Eqn (4.2.6), i.e.

$$h_{LSCV} = \underset{h}{\text{argmin}}\ \mathbb{E}[\mathcal{L}_{psl}(\hat{f}, Y)]_{emp} \tag{4.2.7}$$

However, this method caused a high variability ([89]) and may lead to a bandwidth with a high variance ([**?**]).

LOOCV MISE of CDF: The LOOCV for estimating MISE of CDF was proposed by [90] and [63] but from different point of view. Firstly, the LOOCV for estimating the MISE of $\hat{F}$ is by substituting IBL from Eqn (3.3.13) into $\mathcal{L}$ in Algorithm 8. This results to [63] empirical generalization IBL is

$$\mathbb{E}_{emp}[\mathcal{L}_{ibl}(\hat{F}_{-i}, Y)] = \frac{1}{N}\sum_{i=1}^{N}\int \Big(H(t - Y_i) - \hat{F}_{-i}(t)\Big)^2 \, dt. \tag{4.2.8}$$

[63] showed that the expectation of the difference of $\mathbb{E}_{emp}[\mathcal{L}_{ibl}(\hat{F}_{-i}, Y)]$ and $\mathbb{E}_{emp}[\mathcal{L}_{ibl}(F, Y)]$ leads to the MISE$[\hat{F}]$. The optimal bandwidth is obtained by minimising Eqn (4.2.8) w.r.t h,

$$h_{LOOCVF} = \underset{h}{\text{argmin}}\ \mathbb{E}_{emp}[\mathcal{L}_{ibl}(\hat{F}_{-i}, Y)]. \tag{4.2.9}$$

Another LOOCV estimator to estimate weighted MISE was proposed by [90]. Since the method is based on discrete error MISE, we will not explain it further.

4.2.2 Estimation of Asymptotic MISE

This section is to provide a review estimating the bandwidth by the asymptotic MISE. The asymptotic MISE arise from the limiting property by expanding the MISE using Taylor expansion and taking some limits for the MISE to approach 0. However, the asymptotic MISE (AMISE) also depends on the true distribution. Therefore, different methods were proposed to estimate AMISE. In this section, we describe the asymptotic MISE for PDF and CDF and compare different methods of estimating bandwidth using them.

Asymptotic AMISE for PDF

Recall the MISE$[\hat{f}]$ of kernel PDF from Eqn (4.1.6). The asymptotic MISE, AMISE$[\hat{f}]$ when $N \to \infty$ ([91]) is

$$\text{AMISE}[\hat{f}] = \frac{h^4\kappa_2^2}{4}R(f'') + \frac{1}{Nh}R(K). \tag{4.2.10}$$

However, AMISE$[\hat{f}]$ still depends on the true unknown PDF via the term $R(f'')$. Therefore, it still needs to be estimated where the estimator of AMISE$[\hat{f}]$ is shown in Eqn (4.2.11)

$$\psi(h) = \frac{h^4\kappa_2^2\hat{R}(f'')}{4} + \frac{1}{Nh}R(K). \tag{4.2.11}$$

Optimal bandwidth is then estimated by minimising Eqn (4.2.11), is

$$\hat{h} = \left(\frac{R(K)}{N\kappa_2^2 \hat{R}(f'')} \right)^{-1/5} \tag{4.2.12}$$

[5], [89], [69], [67], [64], [68] and [92] have addressed methods to estimate the term $\hat{R}(f'')$. Below are some methods that estimate AMISE$[\hat{f}]$. We categorized them into 5 groups.

Parametric Assumption: In this method, $\hat{R}(f'')$ is replaced with its parametric distribution. [5] replaced the unknown $\hat{R}(f'')$ with the $R(f'')$ of a Normal distribution.

The estimated bandwidth obtained by minimising the estimator is

$$\hat{h} = 1.06\sigma N^{-1/5}. \tag{4.2.13}$$

However, this only works when the true distribution is Normal. [5] also proposed to use Silverman's rule of thumb (ROT),

$$\hat{h}_{ROT} = 0.9AN^{-1/5} \tag{4.2.14}$$

where A = min $\left(\sigma, \frac{IQR}{1.34}\right)$, σ and IQR are the standard deviation and inter-quantile range of the sample, respectively. The reduction from 1.06 to 0.9 ensures any bi-modality is not missed.

Biased cross-validation: This method was proposed by [93] and was motivated by [94] by replacing $\hat{R}(f'')$ with,

$$\hat{R}(f'') = \hat{R}(\hat{f}'') - \frac{R(K)}{N^2 h^5}. \tag{4.2.15}$$

where $\hat{R}(\hat{f}) = \int \hat{f}''\ dx$ ([87]). Then, applying the LOOCV, the estimator for AMISE is reduced to

$$\psi_{BCV}(h) = \frac{R(K)}{Nh} + \frac{\sigma^2}{2hN^2} \sum_{j \neq i}^{N} \sum_{i=1}^{N} K_h'' * K_h''(Y_i - Y_j). \tag{4.2.16}$$

The bandwidth from this method is $\hat{h}_{BCV}$ is obtained by minimising Eqn (4.2.16), i.e.,

$$\hat{h}_{BCV} = \underset{h}{\operatorname{argmin}}\, \psi_{BCV}(h) \tag{4.2.17}$$

It is found that this method removes the problem of multiple minima when compared method proposed by [62] and [65].

Maximal smoothing: [95] and [89] proposed to use a method that results to an over-smooth estimated PDF. [95] proposed boundaries for kernel estimators and stated that estimated bandwidth near the bounds will lead to an optimal distribution. [89] proposed to replace $R(f'')$ with a scale function, for example standard deviation. This method was proposed because using cross-validation approach to estimate MISE result to a high variability (i.e. the statistical information varies due to the repeated sampling of cross-validation even when the data is from the same distribution). This method removes unwanted features from using a smaller bandwidth causing the estimated distribution is less flexible and overly smooth. The drawback from using this method is the lost of information due to the maximum smoothing ([89]) by using a scale function. When using the sample standard deviation, s, the bandwidth from maximal smoothing principle is

$$\hat{h}_{max} = 3 \times (35)^{-1/5} s \left(\int K''(u)\, du \right)^{-1/5} N^{-1/5} \tag{4.2.18}$$

Multi-stage: This method is almost similar to biased cross-validation ([93]). However, the estimate of $R(f'')$ of Eqn (4.2.11) is $\hat{R}(f''_a)$ which depends on a different bandwidth, a. Note that a and h are two different bandwidth, in which the former is used to estimate $R(f''_a)$ while the latter is used for estimating the entire distribution. The estimator of AMISE$[\hat{f}]$ is

$$\psi_{data}(h) = \frac{h^4 \kappa_2^2 \hat{R}(f''_a)}{4} + \frac{1}{Nh} R(K). \tag{4.2.19}$$

To find $\hat{R}(f''_a)$, the bandwidth a needs to be estimated. Then, using a to estimate $\hat{R}(f'')$ which is latter used to estimate AMISE$[\hat{f}]$. [68] and [67] proposed multi-stage method to estimate $\hat{R}(f''_a)$ and later used to find $\hat{h}$.

i. [68] proposed to estimate a using

$$\hat{a}_{SJ} = \left(\frac{2L^{(4)}(0)}{N(\kappa_L)_2^2}\right)^{1/7} R(f''')^{-1/7} \tag{4.2.20}$$

However, $\hat{a}_{SJ}$ has the term $R(f''')$ which needs to be estimated. The 1-step method to estimate the bandwidth is as follows:

(a) Firstly, $R(f''')$ can be estimated using a scaled Normal distribution where we define the estimated $R(f''')$ as $\hat{R}^N(f''')$

(b) Second, use $\hat{R}^N(f''')$ to estimate $\hat{a}_{SJ}$ of Eqn (4.2.20)

(c) Then, using $\hat{a}_{SJ}$ to estimate $\hat{R}(f''_a)$ using the Eqn (4.2.21)

$$\hat{R}_{SJ}(f''_a) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} L^{(4)} \left(\frac{Y_i - Y_j}{\hat{a}_{SJ}}\right) \tag{4.2.21}$$

where e $L^{(4)}$ is the kernel function that is 4 time differentiable and not necessarily the same as K.

(d) Finally, the estimated h is

$$\hat{h}_{SJ} = \left(\frac{R(K)}{N\kappa_2^2 \hat{R}_{SJ}(f''_{\hat{a}})}\right)^{1/5}. \tag{4.2.22}$$

ii. [67] and [68] proposed another method to estimate a which is a function of h. The method proposed by [67] follows the same step (i.a) to estimate $R(f)$ and $R(f''')$ which is by substituting f with a scaled Normal distribution which is used to compute $\hat{a}_{PM}(h)$,

$$\hat{a}_{PM}(h) = \left(\frac{18R(L^{(4)})}{(\kappa_L)_2^2}\right)^{1/13} \left(\frac{R(f)}{R^2(f''')}\right)^{1/13} \frac{h^{10/13}}{N^{-2/3}} \tag{4.2.23}$$

where $L^{(4)}$ is the kernel function that is 4 time differentiable and L a symmetric kernel that is not necessarily the same kernel as K used to estimate PDF. $\hat{a}_{PM}(h)$ is used to estimate $\hat{R}(f''_a)$ to obtain Eqn (4.2.24),

$$\hat{R}_{PM}(f''_{\hat{a}(h)}) = \frac{1}{N(N-1)} \sum_{i \neq j}^{N} L^{(4)} \left(\frac{Y_i - Y_j}{\hat{a}_{PM}(h)}\right). \tag{4.2.24}$$

In Eqn (4.2.24), the diagonal elements are not included when computing the sum. The estimated bandwidth via this method is

$$\hat{h}_{PM} = \left(\frac{R(K)}{N\kappa_2^2 \hat{R}_{PM}(f''_{\hat{a}(h)})} \right)^{1/5}. \tag{4.2.25}$$

In addition, [68] proposed another method motivated by [67] and [96] but uses Eqn (4.2.26) to estimate a,

$$\hat{a}_{SJ}(h) = \left(\frac{2L^{(4)}(0)}{N(\kappa_L)_2^2} \right)^{1/7} R(f''')^{-1/7} h^{5/7} \tag{4.2.26}$$

where $a_{\hat{SJ}}(h)$ is a function of h and used to compute $\hat{R}_{SJ}(f''_a)$ and later the bandwidth as in Eqn (4.2.27),

$$\hat{h}_{SJ1} = \left(\frac{R(K)}{N\kappa_2^2 \hat{R}_{SJ}(f''_{a_{S\hat{J}}(h)})} \right)^{1/5}. \tag{4.2.27}$$

[68] proposed another method to estimate h by using $\hat{R}_{SJ}(f''_{a_{\hat{SJ}}(h)})$ to minimise Eqn (4.2.19). The estimated bandwidth is

$$\hat{h}_{SJ2} = \underset{h}{\operatorname{argmin}} \frac{h^4 \kappa_2^2 \hat{R}_{SJ}(f''_{a_{S\hat{J}}(h)})}{4} + \frac{1}{Nh} R(K) \tag{4.2.28}$$

[67] and [68] proposed a different equations Eqn (4.2.24) and Eqn (4.2.21) to estimate $\hat{R}(f''_a)$ where the latter suggested to include the diagonal elements in the summation. This cause an additional bias but is cancelled out with the negative bias from smoothing. $\hat{h}_{SJ1}$ is found be be suitable for Gaussian mixture with different mean and variance and for smaller sample while $\hat{h}_{PM}$ by [67] performs better for Normal distribution and Gaussian mixture in the simulation study.

Higher-order kernel: [69] proposed to use a higher order of kernel function by including the next order term from the expansion of Eqn (4.1.6), i.e. $R(f''')$. This method leads to an optimal asymptotic performance. Then, the estimate of AMISE$[\hat{f}]$ is now,

$$\psi_{HO}(h) = \frac{1}{NK} R(K) + \frac{h^4 \kappa_2^2 \hat{R}(f'')}{4} - \frac{h^6 \kappa_2 \kappa_4 R(f''')}{24}. \tag{4.2.29}$$

$\hat{R}_1(f'')$ is the estimate of $R(f'')$ uses a higher order kernel. The estimated bandwidth obtained is

$$\hat{h}_{HO} = \frac{R(K)}{N\kappa_2^2\hat{R}_1(f'')} + \frac{R(K)\hat{R}_1(f''')}{20\kappa_4^2\hat{R}_1(f'')}\left(\frac{R(K)}{N\kappa_2^2\hat{R}_1(f'')N}\right)^{3/5}. \tag{4.2.30}$$

The estimate of $\hat{R}_1(f'')$ and $\hat{R}_1(f'')$ uses the same method as in [68] above.

AMISE for CDF

The AMISE for kernel CDF is,

$$\text{AMISE}[\hat{F}] = \frac{1}{N}\int F(y)(1-F(y))\,dy - \frac{2h}{N}\int yK(y)I(y)\,dy + \frac{1}{4}h^4\kappa_2^2R(f') \tag{4.2.31}$$

where $\kappa_2 = \int x^2K(x)\,dx$, $R(f') = \int f'(x)^2\,dx$, ([9]). The AMISE$[\hat{F}]$ also depends on the unknown term $R(f')$. Data driven approach to estimate the AMISE of CDF Methods to estimate this have discussed by [66], [97] and [70]. Let the estimator of AMISE$[\hat{F}]$ be

$$\Psi(h) = \frac{1}{N}\int F(y)(1-F(y))\,dy - \frac{2h}{N}\int yK(y)I(y)\,dy + \frac{1}{4}h^4\kappa_2^2\hat{R}(f') \tag{4.2.32}$$

where $\hat{R}(f')$ is the estimator for $R(f')$. The bandwidth obtained by minimising Eqn (4.2.32) is

$$\hat{h} = \left(\frac{\rho(K)}{N\kappa_2^2R(f')}\right)^{1/3} \tag{4.2.33}$$

where $\rho(K) = \int xK(x)I(x)\,dx$.

Plug-in: [70] proposed a straightforward method to estimate $R(f')$ by referencing to a distribution. The estimated bandwidth is

$$\hat{h}_{HP} = \frac{s}{N^{1/3}}\left(4\sqrt{\pi}\int 2yK(y)I(y)\,dy\right)^{1/3} \tag{4.2.34}$$

When f is substituted with a Normal distribution, the estimated bandwidth $\hat{h}^N_{HP} = 1.59sN^{-1/2}$, where s is the standard deviation obtained by the data. However, this method is only good estimate if dataset is Normally distributed.

Multi-stage: Several data-driven methods to estimate the bandwidth were also proposed using the AMISE$[\hat{F}]$.

i. [66] claimed that the use of LOOCV to estimate MISE$[\hat{F}]$ is actually a leave-none-out cross-validation and proposed to use data-driven method to estimate $\Psi(h)$ using a weighted AMISE$[\hat{F}]$. The weight function, $W(x)$, that is used is bounded and supported on a compact set ([90]), leading to an estimator

$$\begin{aligned}\Psi_1(h) = &\int F(y)(1-F(y))W(y)f(y)\,dy- \\ &\frac{h}{N}2\int f(y)^2W(y)\,dy\int K(y)I(y)\,dy+ \\ &\frac{h^4}{4}\int \left(f'(y)\right)^2 f(y)W(y)\,dy\left(\int y^2K(y)\,dy\right)^2. \end{aligned} \tag{4.2.35}$$

Unlike Eqn (4.2.31), there are two terms to be estimated in above, $\int f(y)^2W(y)\,dy$ and $\int \left(f'(y)\right)^2 f(y)W(y)\,dy$, we refer them to V_1 and V_2, respectively. The estimators are denoted as $\hat{V}_1$ and $\hat{V}_2$. Therefore, the estimator for AMISE used by [66] is

$$\begin{aligned}\Psi_2(h) = &\int F(y)(1-F(y))W(y)\,dF(y) - \frac{2h}{N}\hat{V}_1\int K(y)I(y)\,dy+ \\ &\frac{h^4}{4}\hat{V}_2\left(y^2K(y)\,dy\right)^2 \end{aligned} \tag{4.2.36}$$

Using method by [98], a different kernel estimator are used to estimate both terms,

$$\hat{V}_2 = \frac{1}{N(N-1)}\sum_{i\neq j}^{N}\frac{1}{b}L_b\left(\frac{Y_i-Y_j}{b}\right)W(Y_i) \tag{4.2.37}$$

$$\hat{V}_2 = \frac{1}{N^3c^4}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}L_c'\left(\frac{Y_i-Y_j}{c}\right)L_c'\left(\frac{Y_i-Y_k}{c}\right)W(Y_i) \tag{4.2.38}$$

where L_c' is the derivative a kernel L_c, which may be different than the kernel K used to estimate $\hat{F}$ with the bandwidth c. L_b is a kernel function may also be different from K with the bandwidth b. By minimising the estimator Ψ_2,

[66] obtained a bandwidth estimator,

$$\hat{h}_{alt} = \left(\frac{0.25\hat{V}_1}{N\hat{V}_2} \right)^{1/3}. \tag{4.2.39}$$

ii. [97] used AMISE$[\hat{F}]$ as in Eqn (4.2.31). This results to only estimating $\hat{R}(f')$ in Eqn (4.2.32). This approach is similar to [68] and [67] where to estimate $\hat{R}(f')$ is by using an estimator that depends on another bandwidth a. Let $\hat{S}_r(a_r)$ be the estimator for $\hat{R}(f')$ where r is the number of stage. The bandwidth of $\hat{S}_r(a_r)$ is

$$\hat{a}_r = \left[\frac{2L^{(r)}(0)}{-N(\kappa_L)_2^2 \hat{S}_{r+2}} \right]^{1/5}. \tag{4.2.40}$$

where $\hat{S}_{r+2}$ needs to be estimated. For clarity, we provide the steps below and simplify by using $r = 1$.

(a) Firstly, for $r = 1$, we need to estimate $\hat{S}_3$ by using Normal distribution. The estimated $\hat{S}_3$ defined by $\hat{S}_3^N$.

(b) Secondly, use $\hat{S}_3^N$ to compute

$$\hat{a}_2 = \left[\frac{2L^{(2)}(0)}{-N(\kappa_L)_2^2 \hat{S}_3} \right]^{1/5}. \tag{4.2.41}$$

(c) Third, estimate $\hat{S}_2(a_2)$ using Eqn (4.2.42)

$$\hat{S}_2(\hat{a}_2) = \frac{1}{N^2\hat{a}^3} \sum_{i=1}^{N} \sum_{j=1}^{N} L^{(2)} \left(\frac{X_i - X_j}{\hat{a}_2} \right) \tag{4.2.42}$$

(d) Finally, the estimated bandwidth is found by using $\hat{S}_2(\hat{a}_2)$ into Eqn (4.2.33) to obtain

$$\hat{h}_{PB} = \left(\frac{\rho(K)}{N\kappa_2^2 \hat{S}_2(a_2)} \right)^{1/3} \tag{4.2.43}$$

Note that, L is a kernel function that is not necessary similar to K. For $r > 1$, (b) and (c) will be repeated until $\hat{S}_2(\hat{a}_2)$ is obtained. The bandwidth $\hat{a}_2$ is computed using Eqn (4.2.40), $\hat{h}_{PB}$ is shown to perform better for datasets that are separated bimodal.

iii. In addition to a plug-in method, [70] also proposed a data-driven method motivated by [96] and using a k repeated integration by parts to estimate $R(f')$ of $\Psi(h)$ which results to an estimate bandwidth in Eqn (4.2.44)

$$\hat{h}_{Hk} = \left(\rho \hat{R}_k\right)^{1/3} \frac{1}{N^{1/3}} \tag{4.2.44}$$

where $\hat{R}_k = \frac{\Gamma(k+\frac{3}{2})}{2\pi\sigma^{2k+3}}$ and σ is estimated using the dataset.

By simulation experiment, [70] showed that this method performed well for $k \geq 4$ when comparing k between 1 to 8 on 9 mixture distribution from [99].

4.3 Discussion

There are two main methods to estimate the bandwidth of kernel distribution estimators, via cross-validation or plug-in methods. Using cross-validation to estimate MISE and later to estimate bandwidth may result to sampling variation. h_{LSCV} is found to perform better for small sample but results in multiple minima that overcame by $\hat{h}_{BCV}$ ([93]). In addition, we reviewed some of the different approaches proposed by estimating AMISE of PDF and CDF. Each methods performs differently for different datasets. For example, h_{ROT} are $\hat{h}_{PM}$ are suitable for Normal distribution. $\hat{h}_{max}$ resulted to an over-smoothed distribution. Therefore, each method's performance depends on the dataset itself.

Chapter 5

Efficient Computation of Loss Functions for Distribution Estimation

5.1 Introduction

The objective of this chapter is to provide a method that efficiently computes and evaluates the probabilistic loss functions: (1) log-loss in Eqn (3.3.10); (2) PSL in Eqn (3.3.11); (3) IBL in Eqn(3.3.13) of kernel mixture distribution at an observation point. We describe 'efficient' in this context as we aim to have a method that is applicable for most symmetric kernel functions. In addition, the method is applicable for standard kernel functions and can be extended to kernel mixtures distribution and other known family of parametric distribution such as the Normal, Logistics and Uniform distributions.

Loss functions are important in the supervised learning task as it is a tool to evaluate the performance of the prediction. In the regression supervised learning, the loss functions are comparing the 'value' of the prediction to the true 'value'. On the other hand, the probabilistic loss functions compare a distribution with the value or events it materialized ([60]). Therefore, to compute the loss functions requires the knowledge of the distribution defining functions (e.g. PDF or CDF).

In Section 3.3.1.5, we described how the divergence of expected generalization loss of the true distribution and the expected generalization loss of the predicted distribution are equal to the criteria functions, MISE and KL-divergence. However, due to the unknown true distribution, the expected generalization loss of the true distribution is also unknown. Therefore, the expected generalization losses (log-loss, PSL, IBL) of the predicted distribution are estimates of the criteria functions. The expected generalization loss functions are estimated by their respective empirical losses . Therefore, the empirical log-loss, PSL and IBL are estimates of KL-divergence and MISE. Due to this chain of relationships, the computation of the loss functions is important and needs to be accurate for parameter selection and evaluation processes. One way to do so is by computing the analytical expression or a closed-form expression of the probabilistic loss functions which is in terms of finite mathematical expression rather than a function. Considering the loss functions are in terms of the predicted distribution, using an efficient method that able to derive the closed-form of the loss functions can be a good solution.

The computation of the log-loss is straight-forward because it only depends on the PDF. However, both PSL and IBL depend on the L2-norm of PDF and CDF, respectively. This complicates the computation because each PDF and CDF are in different form (e.g. the L2-norm of the PDF for Gaussian mixture is different from L2-norm of the PDF of Logistic distribution).

The method proposed to compute the loss function uses the properties of the kernel functions and elementary integration. The functions derived from kernel functions will inherit the properties of the latter. The method we use to compute the loss functions for all kernels is by providing a general computation so that it can be applied to compute the loss functions for all or most of the symmetric kernel estimators. Although we proposed this method for symmetric kernel distribution, the method can be implemented by non-symmetric kernel to a certain point. The method can also be generalized to be used for kernel mixture distributions, Normal, Logistic and Uniform parametric distributions with the right substitution. The PDF and CDF of a kernel distribution will inherit the properties of the kernel functions.

From this method, the analytic expression (closed-form expression) of the probabilistic loss functions for nonparametric kernel-based distribution can be obtained. Hence, using this method we compute the closed-form of the CDF, L2-norm of PDF, L2-norm of CDF and L2-norm of CCDF for 11 symmetric kernels. The L2-norms are found by computing the partial L2-products which can be found in Appendix B.1. We also provide the algorithms for using the derivations for mixture kernel distribution.

This chapter is organized as follow. Firstly, we explain the reasons behind the derivation of analytical expression of the losses. Then, we will provide the steps leading to the analytical expression and the algorithms to compute the mixture kernel based distributions. The exact computations of each kernel will be summarized in Table 5.1 which is linked to Appendix B.1. We will discuss an alternative way of computing the IBL at the end.

5.2 Importance of Analytic Expression of the Loss Functions

Here, we consider the importance of the analytical expression of the probabilistic loss functions: (1) log-loss; (2) PSL; (3) IBL; to evaluate the losses of a distribution at an observation point. These three probabilistic loss functions are linked to the criteria shown in Chapter 3 where

i. the divergence of the expected generalization log-loss of predicted PDF and the expected generalization log-loss of true PDF relates to KL-divergence

ii. the divergence of the expected generalization PSL of the PDF of predicted PDF and the expected generalization PSL of the PDF of true distribution is equal to the MISE of PDF

iii. the divergence of the expected generalization IBL of the CDF of predicted distribution and the expected generalization IBL of the CDF of true distribution is equal to the MISE of CDF.

As discussed in Chapter 3, MISE and KL-divergence can be estimated by the expected generalization loss of the predicted distribution while assuming the expected generalization loss of the true distribution to be constant. The expected generalization loss can be estimated empirically. To estimate the expected generalization loss empirically, the true form of the loss functions is needed. Compared to the deterministic supervised setting where loss functions compare the predicted value and the true value, the probabilistic loss functions compares a distribution defining function with a value. Therefore, the probabilistic loss function depends on the distribution defining function. Analytical expression of the probabilistic loss functions is expected to provide a more accurate calculation of the empirical loss. Furthermore, analytical expression are less time consuming and better for large data. The computation provides a generalized way to compute the loss functions not only for standard kernel distribution but they can reproducible for kernel mixtures distribution.

5.3 Kernels and Mixtures Distribution

In this section, we describe the relationship between kernel and mixture PDF. A kernel function is also a PDF, see Section 2.3. A kernel PDF is a mixture of kernel functions. Here, we explain the relationship between standard kernel and mixture kernel as it is important later to extend the method of computing the probabilistic loss function to the mixture kernel.

A kernel function is defined below in Def 5.3.1.

Definition 5.3.1. *A kernel function, K, is a non-negative function $K : \mathbb{R} \to [0 \cup \mathbb{R}^+]$ that integrates to* 1.
The K is called symmetric when,

$$K(u) = K(-u) \qquad \textit{for all } u \in \mathbb{R}. \tag{5.3.1}$$

In Section 2.3, we mentioned that a kernel function itself is a PDF because it fulfils the requirement for a PDF. Recall a mixture distribution in Def 2.2.3. A mixture kernel distribution is defined as follows.

Definition 5.3.2. *Let $x_1, \ldots, x_N \in \mathbb{R}$ be a vector of observations. The continuous mixture distribution with kernel $K_i : \mathbb{R} \to \mathbb{R}$, observation x_i and weight $w_i > 0$ for $i = 1, \ldots, N$ where $\sum_{i=1}^{N} w_i = 1$ is a distribution with PDF*

$$g(y) = \sum_{i=1}^{N} w_i K_i \left(y - x_i\right) \tag{5.3.2}$$

and CDF

$$G(y) = \sum_{i=1}^{N} w_i I_i \left(y - x_i\right) \tag{5.3.3}$$

where $I_i(y) = \int_{-\infty}^{y} K_i \left(u - x_i\right) \, du$ for $i = 1, \ldots, N$.

Def 5.3.2 is a general expression of mixture kernel PDF and CDF. w_i, $i = 1, \dots, N$ is the weight for each x_i ([100]) and may vary. When $w_i = \frac{1}{N}$ for all $i = 1, \dots, N$, the mixture is called 'uniform mixture'. When w_i, $i = 1, \dots, N$, is not constant, it can be intepreted such that each observation x_i have different amount of information about distribution ([100], [101]). For example, in boosted kernel PDF by [80], the weight is updated in each boosting step.

The proposition below shows how a model of a mixture distribution obtained from a kernel distribution.

Proposition 5.3.1. *Let X_i, $i = 1, \dots, N$ be some random variables with CDF of F_i and F_i is the CDF of a kernel distribution, i.e. $F_i(x) = \int_{\infty}^{x} K_i(u)\, du$. Let I t.v.i $1, \dots, N$. Suppose we have observations $x_1, \dots, x_N$. Then, the random variable $X_I + x_I$ has a mixture distribution with CDF of kernel, F_i, observation x_i and weight $\theta_i = P(I = i)$ for $i = 1, \dots, N$.*

Proof. Let $Y = X_I + x_I$, by elementary calculation

$$\begin{aligned}
&P(Y \leq y|I = i)\\
=&P(X_I + x_I \leq y|I = i)\\
=&P(X_i + x_i - x_i \leq y - x_i|I = i)\\
=&P(X_i \leq y - x_i)\\
=&F_i(y - x_i)
\end{aligned}$$

where $P(X_I \leq y - x_i|I = i)$ is CDF of x_i. By law of total probability, the marginal distribution is

$$\begin{aligned}
G(y) =&P(Y \leq y)\\
=&\sum_{i=1}^{N} P(Y \leq y|I = i)P(I = i)\\
=&\sum_{i=1}^{N} \theta_i F_i(y - x_i)
\end{aligned}$$

and this is the same as in Def 5.3.2 when $\theta_i = \frac{1}{N}$ for all $i = 1, \dots, N$. □

The proposition above shows how to obtained the mixture CDF for vector of random variables. [102] showed the mixture CDF for single random variable.

5.4 Computation of Probabilistic Loss Functions

In this section, we provide a method that efficiently computes the probabilistic loss functions: (1) log-loss; (2) PSL; (3) IBL; for standard kernel distribution and kernel mixture distributions at an observation point. This method uses the properties of the kernel function and the relationship between kernel function and kernel mixture. From this method, the probabilistic loss functions are obtained for standard kernel and mixture kernel distributions.

Based on these probabilistic loss functions there are 4 terms needed: (1) PDF; (2) L2-norm for PDF; (3) L2-norm for CDF; (4) L2-norm for complementary CDF (CCDF). To obtain (3) and (4), we need the CDF.

We divide the computation into two parts. In Section 5.4.1, we compute the functions needed to compute the loss for vanilla (standard) kernel distribution. Later, we generalized into the kernel mixture distributions in Section 5.4.2.

5.4.1 Computation of Loss Functions for Kernel Distribution

Recall the definition of kernel function in Def 5.3.1. In this section, we define the functions CDF, partial L2-product of PDF (to compute the L2-norm of the PDF), partial L2-product of CDF (to compute the L2-norm of CDF) and partial L2-product of CCDF (to compute L2-norm of CCDF) derived from the kernel function K in Def 5.4.1.

Definition 5.4.1. *Let K be a kernel function as in Definition 5.3.1. Then, we define the following notations:*

i. The CDF associated with K is

$$F_K(x) = \int_{-\infty}^{x} K(u)\, du. \tag{5.4.1}$$

ii. The partial L2-product of the PDF associated with K at two different points, 0 and $c \in \mathbb{R}$, until a limit $a \in \mathbb{R} \cup \infty$ is defined as

$$\lambda_K(a, c) = \int_{-\infty}^{a} K(u)K(u - c)\, du. \tag{5.4.2}$$

where $a \in \mathbb{R} \cup \infty$.

iii. The partial L2-product of the CDF, F_K, associated with K at two different points, 0 and $c \in \mathbb{R}$, is defined as

$$\gamma_K(a, c) = \int_{-\infty}^{a} F_K(t) F_K(t - c)\, dt. \tag{5.4.3}$$

where $a \in \mathbb{R}$.

iv. The partial L2-product of the complementary CDF (CCDF) at two different points, 0 and $c \in \mathbb{R}$, is defined as

$$\xi_K(a, c) = \int_{a}^{\infty} (1 - F_K(t))(1 - F_K(t - c))\, dt. \tag{5.4.4}$$

where $a \in \mathbb{R}$.

F_K, λ_K, γ_K and ξ_K are all based on kernel K and inherits the properties of K. For this method, we assume that the partial L2-product λ_K for the kernel functions K exist and are finite. The partial L2-product at two different points 0 and c also means that product of two kernel-based functions in which one is shifted by $c \in \mathbb{R}$. In other words, it is the convolution of two functions after one is shifted by c. Furthermore, Eqn (5.4.2) is a general form in which a can take any value in $\mathbb{R}$ including ∞. The CDF $F_K(x)$ in Eqn (5.4.1) represents the area of the curve $\int_{-\infty}^{x} K(u)\, du$ which is also known to be the probability of less than or equal x. When K is symmetric, the CDF in Eqn (5.4.1) derived from the kernel function has the properties in Lemma 5.4.1.

Lemma 5.4.1. *Let K be a symmetric kernel function in Def 5.3.1 and let the CDF be defined as in Eqn (5.4.1). Then, the following hold.*

i. $\lim_{x \to \infty} F_K(x) = 1$
ii. $\lim_{x \to -\infty} F_K(x) = 0$
iii. $F_K(-x) = 1 - F_K(x)$*, this follows from the symmetric property of* K
iv. $F_K(0) = \frac{1}{2}$

Proof. i. This follows from Def 5.3.1 and Def 5.4.1(i) which will integrate to 1.
ii. This follows from Def 5.4.1(i).

iii. From Def 5.3.1, we have K integrates to 1, where

$$\begin{aligned}\int_{-\infty}^{x} K(u)\,du + \int_{x}^{\infty} K(u)\,du =& 1\\ \int_{x}^{\infty} K(u)\,du =& 1 - \int_{-\infty}^{x} K(u)\,du\\ \int_{x}^{\infty} K(u)\,du =& 1 - F_K(x).\end{aligned} \tag{5.4.5}$$

In addition to that, using the property that K is symmetric,

$$\int_{x}^{\infty} K(u)\,du = -\int_{-x}^{-\infty} K(u)\,du = \int_{-\infty}^{-x} K(u)\,du = F_K(-x). \tag{5.4.6}$$

Hence, we can re-write Eqn 5.4.5 as

$$F_K(-x) = 1 - F_K(x). \tag{5.4.7}$$

iv. This follow from (iii), when $x = 0$.

$$\begin{aligned}\int_{-\infty}^{0} K(u)\,du + \int_{0}^{\infty} K(u)\,du =& 1\\ F_K(0) + F_K(0) =& 1\\ F_K(0) =& \frac{1}{2}.\end{aligned} \tag{5.4.8}$$

□

Furthermore, by Lemma 5.4.1, we can express Eqn (5.4.4) in two different ways as shown in Lemma 5.4.2.

Lemma 5.4.2. *Let K be a symmetric kernel as in Def 5.3.1 with the CDF F_K as in Eqn (5.4.1) and $a \in \mathbb{R} \cup \infty$ and $c \in \mathbb{R}$. Then, we have*

i.

$$\xi_K(a, c) = \int_{a}^{\infty} F_K(-t) F_K(-(t - c))\,dt. \tag{5.4.9}$$

ii.

$$\xi_K(a, c) = \gamma_K(-a, -c). \tag{5.4.10}$$

Proof. i. Recall Eqn (5.4.4) and the symmetric kernel K in Def 5.3.1. Then, from (iii) of Lemma 5.4.1 we can obtain,

$$\xi_K(a,c) = \int_a^\infty F_K(-t)F_K(-(t-c))\,dt. \tag{5.4.11}$$

ii. Recall Eqn (5.4.4) and K is the symmetric kernel in Def 5.3.1. Using the result from (i) above and substituting $u=-t$,

$$\begin{aligned}\xi_K(a,c) &= \int_a^\infty F_K(-t)F_K(-(t-c))\,dt\\ &= \int_{-a}^{-\infty} -F_K(u)F_K(u+c)\,du\\ &= \int_{-\infty}^{-a} F_K(u)F_K(u+c)\,du\\ &=\gamma_K(-a,-c).\end{aligned} \tag{5.4.12}$$

□

Lemma 5.4.1 and Lemma 5.4.2 are applicable to the 11 symmetric kernel in Table 5.1.

For generality, we consider now that the kernel functions are both shifted by b and c, where $b, c \in \mathbb{R}$. Then, we can extend Def 5.4.1 into Lemma 5.4.3 below.

Lemma 5.4.3. *Let K be a symmetric kernel function as in Def 5.3.1, then the following integrals hold for the equalities below.*

i. $\int_{-\infty}^a K(u-b)K(u-c)\,du = \lambda_K(a-b, c-b)$
ii. $\int_{-\infty}^a F_K(t-b)F_K(t-c)\,dt = \gamma_K(a-b, c-b)$
iii. $\int_a^\infty F_K(-(t-b))F_K(-(t-c))\,dt = \xi_K(a-b, -(c-b))$
iv. $\xi_K(a-b,-(c-b)) = \gamma_K(-(a-b), -(c-b))$

Proof. i. Let the partial L2-product of the kernel K at two different points $b, c \in \mathbb{R}$, be

$$\int_{-\infty}^a K(u-b)K(u-c)\,du \tag{5.4.13}$$

where $a \in \mathbb{R} \cup \infty$. Equivalently, we can re-write it as

$$\int_{-\infty}^{a-b} K(u)K(u-(c-b))\, du = \lambda_K(a-b, c-b). \tag{5.4.14}$$

When $a = \infty$, we have a special case,

$$\int_{-\infty}^{\infty} K(u-b)K(u-c)\, du = \lambda_K(c-b). \tag{5.4.15}$$

ii. Let the partial L2-product of the CDF of the kernel, K, at two different starting points $b, c \in \mathbb{R}$ be

$$\int_{-\infty}^{a} F_K(t-b)F_K(t-c)\, dt \tag{5.4.16}$$

where $a \in \mathbb{R}$. Equivalently, we can re-write it as

$$\int_{-\infty}^{a-b} F_K(t)F_K(t-(c-b))\, dt = \gamma_K(a-b, c-b). \tag{5.4.17}$$

iii. Let the partial L2-product of $1 - F_K(t)$ for kernel K with two different central points $b, c \in \mathbb{R}$ be defined as

$$\int_{a}^{\infty} F_K(-(t-b))F_K(-(t-c))\, dt \tag{5.4.18}$$

where $a \in \mathbb{R}$. Equivalently, we can re-write as

$$\int_{a-b}^{\infty} F_K(-t)F_K(-(t-(c-b)))\, dt = \xi_K(a-b, c-b). \tag{5.4.19}$$

iv. The proof follows from extending (ii) of Lemma 5.4.2.

□

Once the terms of the CDF, partial L2-product of PDF, partial L2-product of CDF and partial L2-product of CCDF for kernel functions are defined, they can be used to compute the loss functions in terms of the kernel. The loss functions for kernels are defined in Def 5.4.2.

Definition 5.4.2. *Let K be a kernel function as in Def 5.3.1 with the CDF F_K, par-*

tial L2-product of PDF λ_K, partial L2-product of CDF γ_K and partial L2-product of CCDF ξ_K. Let the log-loss, PSL and IBL be defined in Def 3.3.1, Def 3.3.2 and Def 3.3.3, respectively. Let $a, b, c \in \mathbb{R}$. Then, shifted loss functions are

i. $\mathcal{L}_{ll}(K, u) = -\log K(u)$

ii. $\mathcal{L}_{psl}(K, \lambda_K, c) = -2K(u) + \lambda_K(a = \infty, c - b)$

iii. $\mathcal{L}_{ibl}(\gamma_K, \xi_K, a, c) = \gamma_K(a, c) + \xi_K(a - b, c - b)$.

When K is symmetric, (iii) of Def 5.4.2 can be extended by Lemma 5.4.2 for IBL and the expression of IBL in terms of CDF based kernel, F_K is shown below in Lemma 5.4.4.

Lemma 5.4.4. *Let K be a symmetric kernel function as in Def 5.3.1. Let $a, b, c \in \mathbb{R}$. Then, under the symmetric properties of kernel,*

$$\mathcal{L}_{ibl}(F_K, a, b, c) = \gamma_K(a - b, c - b) + \gamma_K(-(a - b), -(c - b)). \qquad (5.4.20)$$

Proof. Proof follows direct substituting the second term on the RHS of Eqn (iii.) with the result (ii) of Lemma 5.4.2 to obtain Eqn (5.4.20). □

5.4.2 Computation of Loss Functions for Mixture Distribution

The method in Section 5.4.1 can be generalized to a mixture distribution. In this section, we extend the method in Section 5.4.1 to compute the probabilistic loss function: (1) log-loss in Eqn (3.3.10); (2) PSL in Eqn (ii.); (3) IBL in Eqn (2.3.6) for a kernel mixture distribution in Def 5.3.2).

In order to compute these probabilistic loss functions for a kernel mixture distribution, we use Def 5.4.1 to compute the CDF, partial L2-product of PDF, partial L2-product of CDF and partial L2-product of complement CDF (CCDF) of the kernel mixture distribution as in Proposition 5.4.1.

Proposition 5.4.1. *Let K be a kernel function in Def 5.3.1. Let $w_i > 0$ be a weight function where $i = 1, \ldots, N$ and $\sum_{i=1}^{N} w_i = 1$. Let $x_1, \ldots, x_N \in \mathbb{R}$ be a vector of observations and $h \in \mathbb{R}^+$ is the bandwidth. Let p be a PDF of mixture such that*

$$p(x) = \sum_{i=1}^{N} \frac{w_i}{h} K\left(\frac{x - x_i}{h}\right) \qquad (5.4.21)$$

and p satisfy the conditions in Definition 5.3.1. Then, we define following:

i. CDF for mixtures

$$F_p\left(a,x\right)=\sum_{i=1}^{N}w_iF_K\left(\frac{a-x_i}{h}\right) \tag{5.4.22}$$

ii. The partial L2-product for the PDF for mixtures

$$\lambda_p\left(a,c,x\right)=\sum_{i=1}^{N}\sum_{j=1}^{N}w_iw_j\lambda_K\left(\frac{a-x_i}{h},\frac{c+\left(x_j-x_i\right)}{h}\right)\cdot\frac{1}{h} \tag{5.4.23}$$

iii. The partial L2-product for the CDF for mixtures

$$\gamma_p\left(a,c,x\right)=\sum_{i=1}^{N}\sum_{j=1}^{N}w_iw_j\gamma_K\left(\frac{a-x_i}{h},\frac{c+\left(x_j-x_i\right)}{h}\right)\cdot h \tag{5.4.24}$$

iv. The partial L2-product for the 1-CDF (CCDF) for mixtures

$$\xi_p\left(a,c,x\right)=\sum_{i=1}^{N}\sum_{j=1}^{N}w_iw_j\xi_K\left(\frac{a-x_i}{h},\frac{c+\left(x_j-x_i\right)}{h}\right)\cdot h \tag{5.4.25}$$

Proof. Let K be a kernel function with a PDF as in Eqn (5.4.21). We show the relationship between the mixture PDF and the kernels as in Lemma 5.4.3.

i. The CDF for mixtures is defined as

$$F_p\left(a,x\right)=\sum_{i=1}^{N}\int_{-\infty}^{a}w_i\frac{1}{h}K\left(\frac{x-x_i}{h}\right)\,dx.$$

By making a substitution of $u=\frac{x-x_i}{h}$, the RHS of $F_p(a)$ above is,

$$=\sum_{i=1}^{N}w_i\int_{-\infty}^{\frac{a-x_i}{h}}K(u)\,du \tag{5.4.26}$$

and from Lemma 5.4.3, $\int_{-\infty}^{\frac{a-x_i}{h}}K(u)\,du=F_K\left(\frac{a-x_i}{h}\right)$. Hence,

$$F_p\left(a,x\right)=\sum_{i=1}^{N}w_iF_K\left(\frac{a-x_i}{h}\right). \tag{5.4.27}$$

ii. Let the partial L2-product of the mixture PDF for $a \in \mathbb{R}$ be defined as

$$\begin{aligned}\lambda_p(a,c,x) &= \int_{-\infty}^{a} p(x)p(x-c)\,dx\\ &= \int_{-\infty}^{a} w_i w_j \frac{1}{h^2} \sum_{i=1}^{N}\sum_{j=1}^{N} K\left(\frac{x-x_i}{h}\right) K\left(\frac{x-x_j-c}{h}\right)\,dx.\end{aligned}$$

Then, by making a substitution of $u = \frac{x}{h}$, we obtain

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \frac{1}{h^2} \int_{-\infty}^{\frac{a-x_i}{h}} K\left(u - \frac{x_i}{h}\right) K\left(\frac{hu-c-x_j}{h}\right) \cdot h\,du \tag{5.4.28}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \frac{1}{h^2} \int_{-\infty}^{\frac{a-x_i}{h}} K(u)\, K\left(u - \frac{c+x_j}{h}\right) \cdot h\,du \tag{5.4.29}$$

and by Lemma 5.4.3(i), $\int_{-\infty}^{\frac{a-x_i}{h}} K(u)\, K\left(u - \frac{c+(x_j-x_i)}{h}\right)\,du = \lambda_K\left(\frac{a-x_i}{h}, \frac{c}{h} - \frac{x_j-x_i}{h}\right)$ where now $a = \frac{a-x_i}{h}$ and $c = \frac{c}{h} - \frac{x_j-x_i}{h}$, hence

$$\lambda_p(a,c,x) = \frac{1}{h}\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \lambda_K\left(\frac{a-x_i}{h}, \frac{c+(x_j-x_i)}{h}\right). \tag{5.4.30}$$

iii. Let the partial L2-product of the CDF be

$$\begin{aligned}\gamma_p(a,c,x) &= \int_{-\infty}^{a} F_p(x)F_p(x-c)\,dx\\ &= \int_{-\infty}^{a} \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j F_K\left(\frac{x-x_i}{h}\right) F_K\left(\frac{x-x_j-c}{h}\right)\,dx\\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \int_{-\infty}^{a} F_K\left(\frac{x-x_i}{h}\right) F_K\left(\frac{x-x_j-c}{h}\right)\,dx\end{aligned}$$

By making a substitution $t = \frac{x}{h}$, to the RHS, we obtain

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \int_{-\infty}^{\frac{a}{h}} F_K\left(t - \frac{x_i}{h}\right) F_K\left(t - \frac{c+x_j}{h}\right) \cdot h\,dt \tag{5.4.31}$$

and from Lemma 5.4.3(ii), we have

$$\int_{-\infty}^{\frac{a-x_i}{h}} F_K(t)\, F_K\left(t - \frac{c+(x_j-x_i)}{h}\right)\, dt = \gamma_K\left(\frac{a-x_i}{h}, \frac{c+(x_j-x_i)}{h}\right). \tag{5.4.32}$$

Therefore, we have

$$\gamma_p(a,c,x) = h\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \gamma_K\left(\frac{a-x_i}{h}, \frac{c+(x_j-x_i)}{h}\right). \tag{5.4.33}$$

iv. The partial L2-product of the $1-$ CDF(CCDF) of mixtures is

$$\xi_p(a,c,x) = \int_a^{\infty} F_p(-x)F_p(-(x-c)))\, dx \tag{5.4.34}$$

$$= \int_a^{\infty} \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j F_K\left(-\left(\frac{x-x_i}{h}\right)\right) F_K\left(-\left(\frac{x-x_j-c}{h}\right)\right) dx \tag{5.4.35}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \int_a^{\infty} F_K\left(\frac{-x+x_i}{h})\right) F_K\left(\frac{-x+x_j+c}{h}\right) dx. \tag{5.4.36}$$

By making a substitution $t = \frac{x}{h}$, on the RHS in the equation above

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \int_{\frac{a}{h}}^{\infty} F_K\left(-t+\frac{x_i}{h}\right) F_K\left(\frac{-th+x_j+c}{h}\right) \cdot h\, dt \tag{5.4.37}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \int_{\frac{a}{h}}^{\infty} F_K\left(-t+\frac{x_i}{h}\right) F_K\left(-t+\frac{x_j+c}{h}\right) \cdot h\, dt \tag{5.4.38}$$

and from Lemma 5.4.3(iii),

$$\int_{\frac{a-x_i}{h}}^{\infty} F_K(-t)\, F_K\left(-\left(t - \frac{c+x_j-x_i}{h}\right)\right)\, dt = \xi_K\left(\frac{a-x_i}{h}, \frac{c+(x_j-x_i)}{h}\right). \tag{5.4.39}$$

Hence, we have

$$\xi_p(a,c,x) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \xi_K\left(\frac{a-x_i}{h}, \frac{c+(x_j-x_i)}{h}\right) \cdot h. \tag{5.4.40}$$

□

Proposition 5.4.1 are the functions for homogeneous kernel mixtures distribution. To compute in terms of uniform weight homogeneous mixtures, w_i and w_j will need to be substituted with $\frac{1}{N}$ for $i, j = 1, \ldots, N$. When the kernel is symmetric, iv of Proposition 5.4.1 can also be expressed in terms on ξ_p as shown in Lemma 5.4.5.

Lemma 5.4.5. *Let K be a kernel function in Def 5.3.1. Let $w_i > 0$ be a weight function where $i = 1, \ldots, N$ and $\sum_{i=1}^{N} w_i = 1$. Let $x_1, \ldots, x_N \in \mathbb{R}$ be a vector of observations and $h \in \mathbb{R}^+$ is the bandwidth. Let p be a PDF of kernel mixture as in Eqn (5.4.21). Then,*

$$\begin{aligned}\xi_p(a, c, x) =& \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \gamma_k(-a, -c, x). \\ =& \gamma_p(-a, -c, x) \end{aligned} \tag{5.4.41}$$

Proof. The proof follows from Lemma 5.4.2 and substituting K with p in Eqn (5.4.21). □

5.4.2.1 Loss functions for Kernel Mixture Distributions

The probabilistic loss functions for kernel mixture distributions $p(x)$ as in Eqn (5.4.21) are as below.

i. Log-loss:

$$\mathcal{L}_{ll}(p, x) = -\log p(x). \tag{5.4.42}$$

ii. PSL:

$$\mathcal{L}_{psl}(p, \lambda_p, c, x) = -2p(x) + \lambda_p(c). \tag{5.4.43}$$

iii. IBL:

$$\mathcal{L}_{ibl}(\gamma_p, \xi_p, a, c, x) = \gamma_p(a, c, x) + \xi_p(a, c, x). \tag{5.4.44}$$

Proposition 5.4.2 shows the substitution of result from Proposition 5.4.1 into the loss functions for kernel mixture distribution.

Proposition 5.4.2. *Let K be a symmetric kernel function as in Def 5.3.1. Let $w_i \geq 0$ be a weight function where $i = 1, \ldots, N$ and $\sum_{i=1}^{N} w_i = 1$. Let $x_1, \ldots, x_N \in \mathbb{R}$ be* *some observations and $h \in \mathbb{R}^+$ is the bandwidth. Let p be a PDF of mixture as in Eqn (5.4.21) and the CDF as in Eqn (5.4.22). The (1) log-loss; (2) PSL; (3) IBL for kernel mixtures are*

$$\mathcal{L}_{ll}(p, x) = -\log\left(\sum_{i=1}^{N} \frac{w_i}{h} K\left(\frac{x - x_i}{h}\right)\right) \tag{5.4.45}$$

$$\mathcal{L}_{psl}(p, \lambda_p, x) = -2\sum_{i=1}^{N} \frac{w_i}{h} K\left(\frac{x - x_i}{h}\right) + \frac{1}{h}\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \lambda_K \left(\frac{x_j - x_i}{h}\right) \tag{5.4.46}$$

$$\begin{aligned}\mathcal{L}_{ibl}(\gamma_p, x) =& h\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(\gamma_K\left(\frac{x - x_i}{h}, \frac{x_j - x_i}{h}\right) + \right.\\ &\left.\gamma_K\left(\frac{-(x - x_i)}{h}, \frac{-(x_j - x_i)}{h}\right)\right).\end{aligned} \tag{5.4.47}$$

Proof. The proof is by directly substituting results from Proposition 5.4.1 to Eqn (5.4.42), (5.4.43) and (5.4.44) with $c = 0$. □

5.4.3 Analytical Expression and Algorithms

In this section, we provide a summary table that list all the partial L2-product kernel based functions and linked to Appendix B.1. The algorithms to compute the loss functions using the derivation listed in Table 5.1 are discussed in 5.4.3.2.

5.4.3.1 Summary Table of Analytical Expression

This section is to provide a summary table on the functions derived from kernel functions. The derivation of the functions in Def 5.4.1 are shown in Table 5.1. Note that we do not derive in terms of mixtures because we want to ensure that the functions derived can be used in multiform of homogeneous mixtures and single distributions. The actual derivations can found in Appendix B.1.

Kernel Name	Kernel, $K(x)$	$F_K(x)$	$\lambda_K(a = \infty, c)$	$\lambda_K(a, c)$	$\gamma_K(a, c)$	$\gamma_K(-a, -c)$
Uniform	B.2.1	B.2.1	B.2.2	B.2.3	B.2.4	B.2.5
Epanechnikov	B.3.1	B.3.1	B.3.2	B.3.3	B.3.4	B.3.5
Quartic	B.4.1	B.4.1	B.4.2	B.4.3	B.4.4	B.4.5
Triweight	B.5.1	B.5.1	B.5.2	B.5.3	B.5.1	B.5.4
Triangle	B.6.1	B.6.1	B.6.2	B.6.3	B.6.4	B.6.5
Tricube	B.7.1	B.7.1	B.7.2	B.7.3	B.7.4	B.7.5
Logistic	B.8.1	B.8.1	B.8.2	B.8.3	B.8.4	B.8.5
Gaussian	B.9.1	B.9.1	B.9.2	B.9.3		
Sigmoid	B.10.1	B.10.1	B.10.2	B.10.3		
Cosine	B.11.1	B.11.1	B.11.2	B.11.3	B.11.4	B.11.5
Silverman	B.12.1	B.12.1	B.12.2	B.12.3	B.12.4	B.12.5

Table 5.1: Table of summary of functions derived by kernel functions.

Note that, for Gaussian and Sigmoid kernels, the last two columns (partial L2-product CDF and partial L2-product CCDF) are empty. This is because there are no closed form for the both expressions.

5.4.3.2 Algorithms

This section focusses on presenting the algorithms to use any of the derived functions from Table 5.1 and Appendix B.1 to compute the loss functions for kernel mixture distribution and also the CDF for kernel mixture. The functions derived in Table 5.1 and Appendix B.1 are in the standard kernel functions. The algorithms in the following are for homogeneous kernel mixtures.

i. Log-loss for kernel mixture in Algorithm 9
ii. PSL for kernel mixture in Algorithm 10
iii. IBL for kernel mixture in Algorithm 11
iv. Mixture CDF in Algorithm 12

Algorithm 9 is the algorithm to compute log-loss for homogenous kernel mixture.

Algorithm 9 Algorithm of Log-loss for kernel mixture PDF

```
Inputs: Data: $x_1, \dots, x_N$ where $x_i \in \mathbb{R}$; Weight functions: $w_1, \dots, w_N$ such that $\sum_{i=1}^{N} w_i = 1$; Kernel distribution, $d$; Kernel function for distribution of type $d.K_h : \mathbb{R} \to \mathbb{R}^+$ with bandwidth $h \in \mathbb{R}^+$. The kernel function is also a mixture component; a value $x \in \mathbb{R}$.
Output: Value the log-loss of the kernel mixture, $\mathcal{L}_{ll}(p, x)$.
Steps:
Define the distribution $d$
Compute the kernel PDF at $x$, $p(x) = \sum_{i=1}^{N} w_i d.K_h(x - x_i)$ as in Eqn (5.4.21)
Compute the log-loss $\mathcal{L}_{ll}(p, x) = -\log p(x)$ as in Eqn (5.4.42).
```

From Algorithm 9, $p(x)$ is the PDF as in Eqn (5.4.21) of a distribution d. By substituting K to a Gaussian kernel, and substituting the observed data to a vector of mean (i.e. $\mu_1, \dots, \mu_N$) with the same standard deviation h, this will result in a univariate Gaussian mixture PDF with weight, w_i. Whereas, instead of using a vector of mean, by substituting a scalar, this will result to univariate Gaussian/Normal PDF.

The Algorithm 10 uses a bandwidth to compute the PSL for kernel mixture at the point $x \in \mathbb{R}$ while Algorithm 11 is used to compute the IBL for kernel mixture distribution using the partial L2-product of CDF and partial L2-product for CCDF, respectively.

Algorithm 10 Algorithm of PSL for kernel mixture

```
Inputs: Data: $x_1, \dots, x_N$ where $x_i \in \mathbb{R}$; Weight function: $w_1, \dots, w_N$ where $\sum_{i=1}^{N} w_i = 1$; Kernel distribution, $d$; Kernel function for distribution $d$, $d.K_h : \mathbb{R} \to \mathbb{R}^+$; The partial L2-product for kernel distribution $d$, $d.\lambda_{K_h} : \mathbb{R} \to \mathbb{R}$; $x \in \mathbb{R}$.
Output: Value of the PSL for kernel mixture $\mathcal{L}_{psl}(p, c, x)$.
Steps:
Define the distribution $d$
Compute the kernel PDF at $x$, $p(x) = \sum_{i=1}^{N} w_i d.K_h(x - x_i)$ as in Eqn (2.2.4)
Compute the partial L2-product of kernel mixture, $\lambda_p(c) = \frac{1}{h} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j d.\lambda_{K_h}(c = x_i - x_j)$ as in Eqn (5.4.23)
Compute the PSL for kernel mixture, $\mathcal{L}_{psl}(p, c, x) = -2p(x) + \lambda_p(c)$ as in Eqn (5.4.43.)
```

Algorithm 11 Algorithms of IBL for kernel mixture

```
Inputs: Data: $x_1, \dots, x_N$ where $x_i \in \mathbb{R}$; Weight function: $w_1, \dots, w_N$ where $\sum_{i=1}^{N} w_i = 1$; Kernel distribution $d$ with CDF $F$; The partial L2-product of CDF of kernel distribution $d$ of type $d.\gamma : \mathbb{R} \to \mathbb{R}$; Upper limit $a \in \mathbb{R}$.
Outputs: A value of the IBL for kernel mixture at the point $a \in \mathbb{R}$, $\mathcal{L}_{ibl}(\gamma_{K_h}, a, c)$
Steps:
Define a kernel distribution, $d$
Compute the partial L2-product of CDF for kernel distribution $d$, $\gamma(a,c) = h \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j d.\gamma(a - x_i, c = x_i - x_j)$ as in Eqn (5.4.24)
Compute the partial L2-product of CCDF for kernel distribution $d$, $\gamma(a,c) = h \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j d.\gamma(-(a - x_i), -(c = x_i - x_j))$ as shown in Lemma 5.4.5.
Compute the IBL for kernel mixture distribution $d$, $\mathcal{L}_{ibl}(\gamma, a, c) = \gamma(a,c) + \gamma(-a,-c)$
```

Algorithm 12 provides the algorithm to compute of homoegeneous CDF and the special case of univariate CDF via kernel method.

Algorithm 12 Algorithm for kernel mixture CDF

```
Inputs: Data: $x_1, \dots, x_N$ where $x_i \in \mathbb{R}$; Weight functions: $w_1, \dots, w_N$ where $\sum_{i=1}^{N} w_i = 1$; Kernel distribution, $d$; CDF of kernel function of type $d.F_{K_h} : \mathbb{R} \to [0,1]$ where $h \in \mathbb{R}^+$ is the bandwidth; $x \in \mathbb{R}$.
Output: A value of a kernel mixture CDF at $x$, $F(x)$
Steps:
Compute $F(x) = \sum_{i=1}^{N} w_i d.F_{K_h}(x - x_i)$
```

5.5 Alternative computation of IBL

In this section, we discuss alternative methods to compute the IBL for evaluating a distribution at a point. These alternative methods are useful when there is no closed-form for the partial L2-product of CDF and the partial L2-product of CCDF of a distribution. In this section, we discuss some methods from multiple literatures to compute IBL to evaluate the kernel distribution at a point.

IBL can exist in many forms. The IBL is usually expressed in the quadrature rule as discussed in [103], [104], [32], [105], [106] and others. The IBL in quadrature form is defined below.

Definition 5.5.1. *Let Y be a random variable of distribution with the CDF F with finite first moment. Let Y' be a copy of the random variable Y. The IBL of F at a point $y \in \mathbb{R}$ is*

$$\mathcal{L}_{ibl}(F, y) = \mathbb{E}|Y - y| - \frac{1}{2}\mathbb{E}|Y - Y'|. \tag{5.5.1}$$

The drawback from using the above expression to compute IBL at a point is it does not use the CDF but using the expectation of a random variable.

Another method to express IBL is shown Proposition 5.5.1. This derivation follows from [107], [108] and [60], such that the second term on the RHS of Eqn (5.5.1) is

$$\frac{1}{2}\mathbb{E}|Y - Y'| = \int_{-\infty}^{\infty} F(t)(1 - F(t))\, dt.$$

Proposition 5.5.1. *Let Y be a random variable where Y t.v.i $\mathbb{R}$. Let K be a kernel function and F_K is its CDF. From Def 3.3.3 and [60], the IBL for F_K is*

$$\mathcal{L}_{ibl}(F_K, y) = \int_{-\infty}^{\infty} |F_K(t) - H(t - y)|\, dt - \int_{\infty}^{\infty} F_K(t)(1 - F_K(t))\, dt. \tag{5.5.2}$$

Proof. Suppose F_K is a CDF for a kernel K that represents the distribution of random variable Y. Then, the IBL of F_K of Y, can be defined using Def 3.3.3,

$$\mathcal{L}_{ibl}(F_K, y) = \int_{-\infty}^{y} F_K(t)^2\, dt + \int_{y}^{\infty} (1 - 2F_K(t) + F_K(t)^2)\, dt. \tag{5.5.3}$$

From [60], the IBL can be written as

$$\mathcal{L}_{gne}(F_K, y) = \mathbb{E}|Y - y| - \frac{1}{2}\mathbb{E}|Y - Y'|$$

where $\frac{1}{2}\mathbb{E}|Y - Y'| = \int_{-\infty}^{\infty} F_K(t)(1 - F_K(t))\, dt$, leading to,

$$\mathcal{L}_{gne}(F_K, y) = \mathbb{E}|Y - y| - \int_{-\infty}^{\infty} F_K(t)(1 - F_K(t))\, dt. \tag{5.5.4}$$

Comparing Eqn (5.5.3) with Eqn (5.5.4) leads to

$$\begin{aligned}\mathcal{L}_{ibl}(F_K, y) &= \int_{-\infty}^{y} F_K(t)\,dt + \int_{y}^{\infty} (1 - F_K(t))\,dt - \int_{-\infty}^{\infty} F_K(t)(1 - F_K(t))\,dt \\ &= \int_{-\infty}^{\infty} |F_K(t) - H(t-y)|\,dt - \int_{-\infty}^{\infty} F_K(t)(1 - F_K(t))\,dt \qquad (5.5.5)\end{aligned}$$

where $\int_{-\infty}^{\infty} F_K(t)(1-F_K(t))\,dt$ is the expectation of absolute difference of random variable Y and Y'. $\int_{-\infty}^{\infty} |F_K(t) - H(t-y)|\,dt$ is the expected value of the absolute error ([109]). □

Using Proposition 5.5.1, for IBL of Gaussian kernel or standard Gaussian distribution $N(0,1)$, the first and second term of Eqn (5.5.1) are

i. $\mathbb{E}|Y - y| = \int_{-\infty}^{y} F_K(t)\,dt + \int_{y}^{\infty} 1 - F_K(t)\,dt = y\,\mathrm{erf}\left(\frac{y}{\sqrt{2}}\right) + \dfrac{2\mathrm{e}^{-\frac{y^2}{2}}}{\sqrt{2}\sqrt{\pi}}$

ii. $\mathbb{E}|Y - Y'| = \frac{1}{\sqrt{\pi}}$

Therefore, the IBL for standard Gaussian distribution is

$$\mathcal{L}_{ibl}(F, y) = y\,\mathrm{erf}\left(\frac{y}{\sqrt{2}}\right) + \frac{2\mathrm{e}^{-\frac{y^2}{2}}}{\sqrt{2}\sqrt{\pi}} - \frac{1}{\sqrt{\pi}} \qquad (5.5.6)$$

as shown in [60], [109] and [103]. [109] derived the IBL for Gaussian mixture using as shown below.

Example 5.5.1. *Let $Y, Y_1, \ldots, Y_N$ t.v.i $\mathbb{R}$ be independent random variables distributed by a CDF F_p where p be a Gaussian mixture PDF. Let $y, y_1, \ldots, y_N$ be the realization of $Y, Y_1, \ldots, Y_N$ and $h \in \mathbb{R}^+$ is the bandwidth. The IBL for Gaussian mixture CDF is*

$$\mathcal{L}_{ibl}(F_p, y) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y-y_i}{2}\left(\mathrm{erf}\left(\frac{y-y_i}{h\sqrt{2}}\right)+1\right) + \frac{h}{\sqrt{2\pi}}\mathrm{e}^{-\frac{(y-y_i)^2}{2h^2}}\right) - \qquad (5.5.7)$$

$$\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{\sqrt{Nh^2}}\frac{y_i-y_j}{2}\,\mathrm{erf}\left(\frac{y_i-y_j}{\sqrt{2}}\right) + \frac{\sqrt{Nh^2}}{\sqrt{2\pi}}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{e}^{-\frac{(y_i-y_j)^2}{2\sqrt{Nh^2}}}\right). \qquad (5.5.8)$$

where

$$\mathbb{E}|Y - Y'| = \sum_{i=1}^{N}\sum_{j=1}^{N}(y_i - y_j)\,\mathrm{erf}\left(\frac{y_i - y_j}{\sqrt{2}\sqrt{Nh^2}}\right) + \frac{2\sqrt{Nh^2}}{\sqrt{2\pi}}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{e}^{-\frac{(y_i-y_j)^2}{2\sqrt{Nh^2}}}. \tag{5.5.9}$$

5.6 Discussion & Conclusion

The objective of this chapter is to provide an efficient computation of the probabilistic loss functions (i.e. log-loss, PSL and IBL) for kernel based distribution estimator. This computation is more useful for IBL and PSL because both rely on the L2-norm of CDF and the L2-norm of PDF, respectively.The methods proposed in Section 5.4 are exploitable for mixture distribution rather focussing on one type of distribution. We summarize below.

i. The computation of log-loss for distribution is straight forward. This makes log-loss is the comfortable choice of evaluation for users without having to compute anything further.

ii. The PSL is the main choice of loss function for distribution estimation using PDF as it measures the bias-variance trade-off. One of the downfall for kernel mixture distribution is the partial L2-product of each kernel PDF is different. Some of the kernels have bounded support which need to be considered for computation.

iii. The use of IBL for distribution estimation has been increasing. For most kernels, the proposed method in Section 5.4 can be seen as a general approach when dealing with symmetric kernel. The method proposed in Section 5.4 is not applicable for Gaussian and Sigmoid kernel. In this the case where the method in Section 5.4 are not suitable, the quadrature approach can be used.

The partial L2-product s of PDF and the partial L2-product s of CDF have been derived (Appendox B.1) and we integrate the functions into **R** package **distr6**.

Chapter 6

Investigation of In-sample and Out-of-Sample Tuning for Distribution Estimation

6.1 Introduction

The objective of this chapter is to investigate the behaviour of the out-of-sample empirical probabilistic loss function for tunings. A loss function is useful for evaluation and for estimating the parameters of a method. For the latter, this is done by minimising the empirical loss with respect to the parameter. For kernel distribution, the bandwidth is an important parameter that determines the output distribution. Therefore, the empirical loss is minimised to estimate bandwidth.

The empirical loss function is an estimate of the prediction loss. There are two ways to compute the empirical loss and we call them: (1) in-sample empirical loss; (2) out-of-sample empirical loss. For distribution estimation, in-sample empirical loss evaluates the estimated PDF (or CDF) on the same dataset it was trained. This is usually called training lost (error). On the other hand, the out-of-sample empirical loss evaluates the estimated PDF (or CDF) on the test set (i.e. a different dataset it was trained). One way to compute the out-of-sample loss is to use cross-validation.

The used of cross-validation for estimating parameters in distribution estimation has been introduced by [57] and [58] using the log-likelihood PDF and maximised it to obtain the optimal bandwidth. [58] explained that leave-one-cross validation (LOOCV) is useful to prevent the limit of the log-likelihood going to infinity as the bandwidth goes to 0 (or negative infinity for log-loss). [62] and [65] later used cross-validation for estimating the MISE of PDF and minimizing the estimated MISE to obtain an optimal bandwidth. PSL is used to estimate the MISE. Even though PSL is advantageous for absolute continuous random variables, real world data are often discretized. Hence, this affects the use of PSL for tuning. [6] stated there exist a threshold that ensure empirical PSL does not lead to negative infinity as the bandwidth goes to 0 ([5]). In this chapter, we investigate the difference in the behaviour of in-sample and out-of-sample empirical log-loss and out-of-sample empirical PSL for tuning.

Firstly, we investigate the difference in the behaviour between in-sample empirical log-loss and out-of-sample empirical log-loss for tuning the bandwidth of Gaussian kernel PDF estimator. To obtain a minimum point, the empirical log-loss is bounded when the parameter goes to 0 and ∞. To show this, we present proofs that compares the use of in-sample method and out-of-sample method during tuning. In this investigation, we use Gaussian kernel PDF estimator and log-loss. The results from the proof show that in-sample empirical log-loss is not bounded. The out-of-sample empirical log-loss for Gaussian kernel PDF is bounded. The results also indicate that for a global minimum point to occur in the out-of-sample method, at least one new data point in the test set (which is not in the training set) is required. The global minimum indicates an optimum bandwidth.

The second investigation is motivated by [6] and [5]. We investigate the behaviour of the out-of-sample empirical PSL to tune the bandwidth for Gaussian kernel PDF. [6] stated that there exists a threshold β to ensure that empirical PSL to works [5]. [5] stated that the threshold relates to a ratio of repeated data points to all data points in a dataset to achieve a minimum point for empirical PSL as the bandwidth is between 0 and infinity. For Gaussian kernel, the threshold $\beta = \frac{1}{2\sqrt{2}-1}$. Our aim is to provide a formal proof from the statement by [5] using a Gaussian kernel PDF estimator when tuning the bandwidth using out-of-sample empirical PSL. We prove that for out-of-sample empirical PSL of a Gaussian kernel PDF to be bounded and achieved a global minimum, the total number of test data points to the number of data points that exist in training and test sets should exceed $2\sqrt{2}$.

Finally, we conduct a simulation experiment to compare: (1) the behaviour of in-sample empirical log-loss with out-of-sample empirical log-loss; (2) the behaviour of in-sample empirical with the out-of-sample empirical PSL; on 6 different datasets. Results of the experiment shows that out-of-sample empirical log-loss is bounded and a minimum point exists whereas the in-sample empirical log-loss is not bounded. The in-sample empirical PSL is not bounded from below. The out-of-sample empirical PSL is not bounded from below for all datasets (i.e. for some datasets, the out-of-sample empirical PSL tends to negative infinity and 0 has the bandwidth tends to 0 and infinity, respectively). Some datasets show the out-of-sample empirical PSL is bounded (i.e. the out-of-sample PSL tends to infinity and 0 as the bandwidth goes to 0 and infinity, respectively). This is because the ratio of total test points to the repeated number of data points in training and test sets is greater than $2\sqrt{2}$. Whereas, datasets in which the out-of-sample empirical PSL does not have a lower bound is because to the ratio of total test points to the repeated number of data points in training and test sets is less than $2\sqrt{2}$.

The rest of the chapter is organized as follows. Section 6.2 is a background on the evaluation and tuning where we explain the regression setting and distribution estimation. Section 6.3 is on investigating the behaviour of in-sample and out-of-sample empirical log-loss for univariate kernel PDF, where we explain the setting and theoretical formulation with conclusion. Section 6.4 discuss and present the results of the proof for out-of-sample empirical PSL for tuning the bandwidth for Gaussian kernel PDF. Section 6.5 will be on a simulation experiment that compares in-sample and out-of-sample empirical loss for log-loss and PSL.

6.2 Tuning and Evaluation

In Section 3.2.4.1, we briefly describe a tuning method which is a part of meta-learning. In this section we review the evaluating and tuning using in-sample and out-of-sample method based on [1] and [2]. The purpose is to understand the difference in the behaviour of the empirical loss functions using the two methods. In the first half of this section, we describe the evaluation and tuning for regression setting as a review before we explain for distribution estimation. We describe the setting, important terms used, explain the difference between in-sample and out-of-sample empirical loss and incorporate the evaluation methods into tuning. In the second half, we describe the evaluation and tuning for distribution estimation.

6.2.1 In-sample vs Out-of-sample Evaluation Method

This section is to present the setting, datasets, learning function, loss function and terms used in this section. Then, we describe the evaluation method and explain the difference between in-sample and out-of-sample evaluation methods.

Let $\mathcal{D} = ((X_1, Y_1), \ldots, (X_N, Y_N)) \sim (X, Y)$ and $\mathcal{D}^* = ((X_1^*, Y_1^*), \ldots, (X_M^*, Y_M^*)) \sim$

(X, Y) be the training and test sets, respectively, where (X, Y) t.v.i $(\mathbb{R}^n, \mathbb{R})$. Let $\hat{f} : \mathbb{R}^n \to \mathbb{R}$ be a function that maps X to Y, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a loss function. We highlight some important terms used throughout this Section 6.2.

i. **In-sample empirical loss:** Evaluates the trained model on the same dataset set it was trained (also known as the training loss / error). The in-sample empirical loss is $\mathbb{E}_{emp}[\mathcal{L}(\hat{f}(X), Y)] = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f(X_i), Y_i)$.

ii. **Out-of-sample empirical loss:** An estimate of the generalization loss which evaluates the trained model on a test set, $\mathcal{D}^*$. The out-of-sample empirical generalization loss is $\mathbb{E}_{emp}[\mathcal{L}(f(X^*), Y^*)] = \frac{1}{M}\sum_{i=j}^{M} \mathcal{L}(f(X_j^*), Y_j^*)$.

Before we explain the tuning methods further, we need to describe in details the evaluation method. In Section 3.2.1.3, we described the importance of the loss function to evaluate the performance of a model. The empirical loss is computed to estimate the generalized loss to oversee the performance of $\hat{f}$ on the population X and Y. In general, the empirical loss is computed by comparing $\hat{f}(X)$ with Y. There are two ways to compute the empirical loss: (1) in-sample; (2) out-of-sample. The difference between in-sample and out-of-sample empirical loss is the use of a test set on the latter.

First, consider the in-sample empirical loss. The learning function $\hat{f}$ is fitted using the train set $\mathcal{D}$ to output a model $\hat{f}(\mathcal{D})$. The model is later used to predict the target variable of the same set $\mathcal{D}$. Then, the model is evaluated by comparing the output of the predicted $\hat{f}(X)$ against Y. The algorithm for computing in-sample empirical loss is shown in Algorithm 13.

Algorithm 13 In-sample Empirical Loss for Regression Setting

```
Inputs:
    Training set: $\mathcal{D} = ((X_1, Y_1), ..., (X_N, Y_N))$; Learning function,
    $\hat{f} : \mathbb{R}^n \to \mathbb{R}$; Loss function, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
Output: In-sample empirical loss, $\mathbb{E}[\mathcal{L}(Y, \hat{Y})]_{emp}$
Steps:
Train $\hat{f}$ on $\mathcal{D}$
Compute $\hat{Y}_i = \hat{f}(X_i)$
Compute $\mathcal{L}(Y_i, \hat{Y}_i)$ for all $i = 1, \ldots, N$
Compute of the average prediction over $N$, $\mathbb{E}[\mathcal{L}(Y, \hat{Y})]_{emp} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(Y, \hat{Y})$
```

However, it is recommended to evaluate the performance of the model on an unseen dataset by computing the out-of-sample empirical loss. Consider the training set $\mathcal{D}$ and test set $\mathcal{D}^*$. The training set $\mathcal{D}$ is fitted to the learning function $\hat{f}$ and output a model. The model uses the features of the test set $\mathcal{D}^*$ to predict the value of the target variable. Then, the out-of-sample empirical loss is computed by comparing the predicted value of the target variable with the actual target variable of the test set, $\mathcal{D}^*$. The algorithm for out-of-sample empirical loss is shown in Algorithm 14.

Algorithm 14 Out-of-sample Empirical Loss for Regression Setting

```
Inputs:
    Training set: $\mathcal{D} = ((X_1, Y_1), ..., (X_N, Y_N))$; Test set:
    $\mathcal{D}^* = ((X_1^*, Y_1^*), ..., (X_N^*, Y_M^*))$; Learning function,
    $\hat{f} : \mathbb{R}^n \to \mathbb{R}$; Loss function, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
Output: The value of out-of-sample empirical loss $\mathbb{E}[\mathcal{L}(Y, \hat{Y})]_{emp}$
Steps:
Train $\hat{f}$ on $\mathcal{D}$
Compute the loss $\mathcal{L}(\hat{f}(X_j^*), Y_j^*)$ for $j = 1, \dots, M$
Compute the average the loss of the prediction over $M$, $\mathbb{E}[\mathcal{L}(Y, \hat{Y})]_{emp} =$
    $\frac{1}{M}\sum_{i=1}^{M} \mathcal{L}(Y_j^*, \hat{Y}_j^*)$
```

6.2.2 In-sample and Out-of-sample Tuning

The purpose of hyperparameter tuning is to select the model (which includes the parameter) that best describe the dataset using a learning function $\hat{f}$. Generally, tuning done by fitting the training set $\mathcal{D}$ on the function $\hat{f}$ for each parameter a_k. This will output a model $\hat{f}_{a_k}(\mathcal{D})$. Each model $\hat{f}_{a_k}(\mathcal{D})$ is evaluated by computing the empirical loss. The model with the minimum empirical loss is selected as the optimal (best) model. Figure 6.1 shows an overview of the tuning stage.

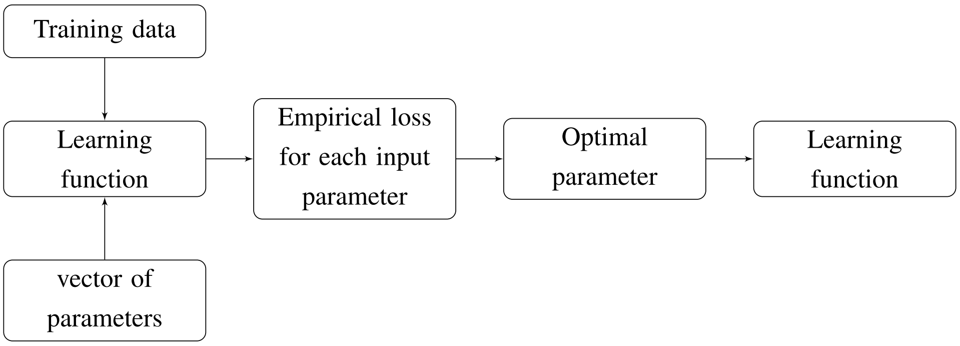


Figure 6.1: Figure of parameter tuning where training set is used for both fitting and tuning.

Tuning algorithm consist of training, predicting, evaluating and minimising the empirical loss. During the evaluating step in tuning, we can either compute the in-sample empirical loss or the out-of-sample empirical loss. However, the former uses the same dataset for training and evaluating and can cause over-fitting where the in-sample empirical loss will tend to decrease as the parameter increase or as the model becomes more flexible. Hence, it will be difficult to select the optimal parameter. In this case, the best parameter will always be the largest value because it resulted the minimum in-sample empirical loss.

To avoid the issue of over-fitting, it is recommended to use a different dataset for training and evaluating, that is by computing the out-of-sample empirical loss in the tuning algorithm. This not only evaluates the model on the unseen dataset but also put a constraint of the empirical loss.

The aim now is to implement Algorithm 14 inside the tuning stage. This is done by further splitting the training set $\mathcal{D}$ into *inner training set*, $\mathcal{T} = ((X_1, Y_1), ..., (X^*_{\tilde{N}}, Y^*_{\tilde{N}})) \overset{i.i.d}{\sim} (X, Y)$ and *inner test set* or validation set, $\mathcal{T}^* = ((X^*_1, Y^*_1), ..., (X_{\tilde{M}}, Y_{\tilde{M}})) \overset{i.i.d}{\sim} (X, Y)$ where (X, Y) t.v.i $(\mathbb{R}^n, \mathbb{R})$. Using the same learning function $\hat{f}$ and the loss function $\mathcal{L}$ with the same vector of parameter $\boldsymbol{a}$, $\mathcal{T}$ is fitted to $\hat{f}$ for each parameter a_k. Then, output model $\hat{f}_{a_k}(\mathcal{D})$ is used to predict the output value of the target variable of $\mathcal{T}^*$. The out-of-sample empirical loss between $\hat{f}_{a_k}(Y^*)$ and Y^* is computed for each f_k. The model $\hat{f}_{a_k}$ that results to the minimum out-of-sample empirical loss is selected as the best model reflecting to the optimal parameter. Algorithm 15 shows the step for out-of-sample tuning for regression setting.

Algorithm 15 Out-of-sample Tuning for Regression Setting

```
Inputs:
    Inner training data: $\mathcal{T} = ((X_1, Y_1), ..., (X_{\tilde{N}}, Y_{\tilde{N}}))$; Inner test data: $\mathcal{T}^* = ((X_1^*, Y_1^*), ..., (X_{\tilde{M}}^*, Y_{\tilde{M}}^*))$; Parameter: $a_1, \ldots, a_K$; Learning function, $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}$; Loss function, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.
Output: The optimal parameter $a_k$
Steps:
for $k = 1, \ldots, K$ do
    Fit $\hat{Y}_i = \hat{f}(X_i)$
    Compute the out-of-sample empirical loss, $\mathbb{E}_{emp}[\mathcal{L}(\hat{f}(X), Y)] = \frac{1}{\tilde{M}} \sum_{j=1}^{M} \mathcal{L}(\hat{f}(X_j^*), \hat{Y}_j^*)$
end for
Select the parameter $a_k$ with the minimum $\mathbb{E}_{emp}[\mathcal{L}(\hat{f}(X), Y)]$
```

6.2.2.1 Tuning & Evaluation

Here, we describe how to put the training stage that involves tuning, prediction and evaluation of the tuned model together.

The tuning algorithm occurs in the training stage. Once the tuned model is obtained, we proceed to the next step, i.e. evaluating the model on test set, $\mathcal{D}^*$. Consider we use out-of-sample for tuning and evaluation and the output of tuning is a tuned model, $\tilde{f}$. This model is used to predict the values of target variable of the test set $\mathcal{D}^*$, $\tilde{f}(X^*)$. The out-of-sample empirical loss is then computed by averaging the loss between $\tilde{f}(X^*)$ and Y^*, i.e. $\frac{1}{M} \sum_{j=1}^{M} \mathcal{L}(\tilde{f}(X_j^*), Y_j^*)$ for all $j = 1, \ldots, M$. We can also use the in-sample tuning and in-sample evaluation (although not recommended). If $\tilde{f}$ is obtained by in-sample tuning, then using the in-sample empirical loss to evaluate the tuned model, we compute $\frac{1}{M} \sum_{i=1}^{N} \mathcal{L}(\tilde{f}(X_i), Y_i)$ for all $i = 1, \ldots, N$. Figure 6.2 is a summary of out-of-sample tuning and evaluation.

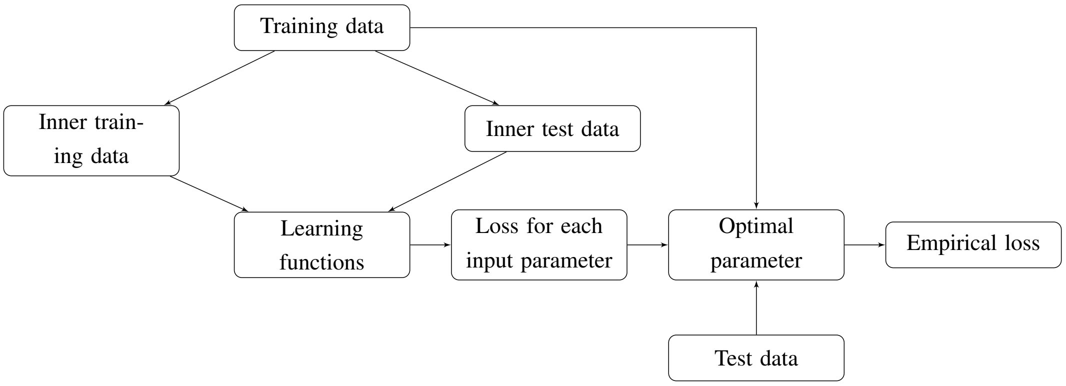
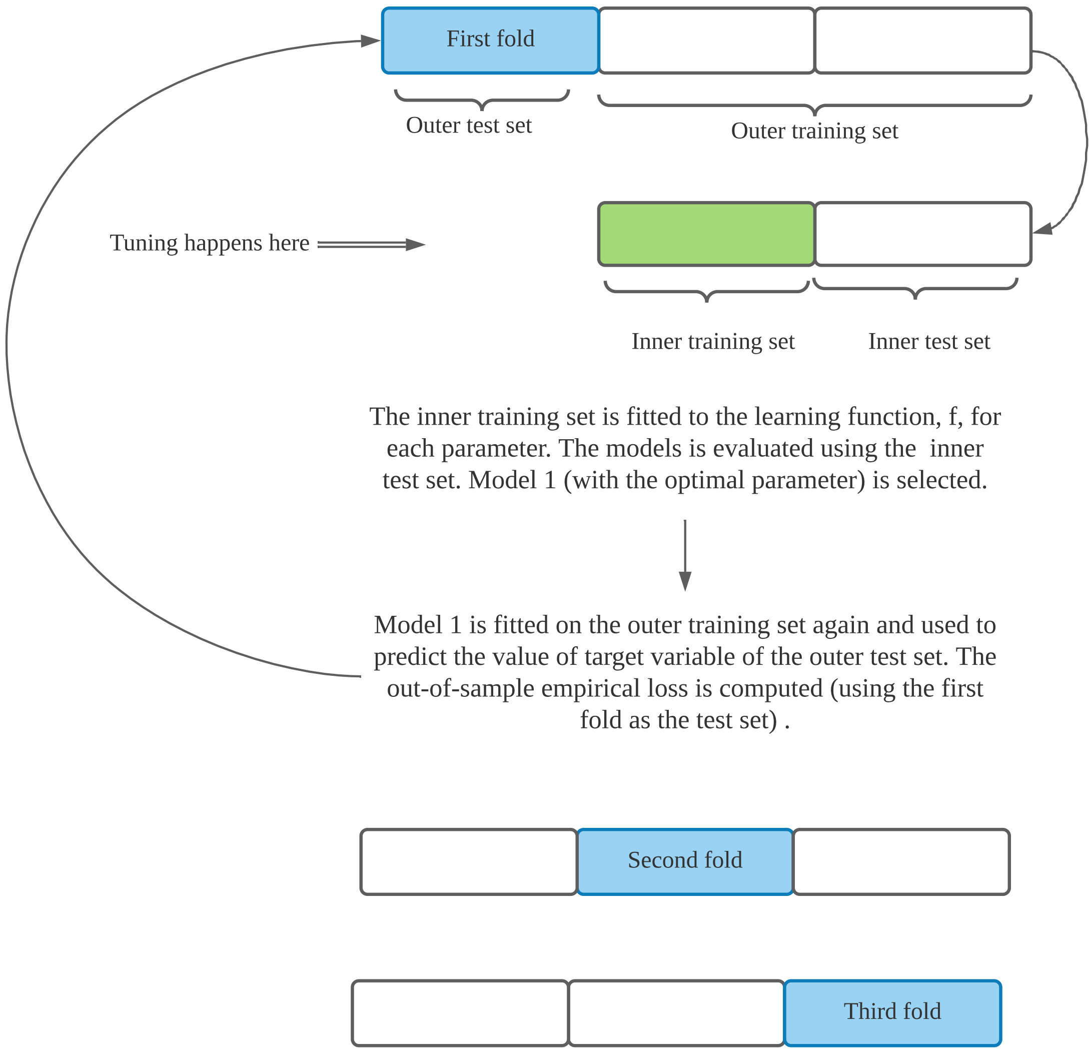


Figure 6.2: Figure of overview an of parameter tuning and evaluation via out-of-sample. The training set is split into inner training and inner test sets. The learning function is fitted on the inner training set and later use to predict the inner test set.

It is not often that we have training set and test set. To overcome this limitation, we can conduct nested resampling method. In this method, we have a dataset in which we have to use for training (using tuning method) and evaluation. Figure 6.3 shows the process of nested resampling via out-of-sample empirical loss for tuning and evaluation while Figure 6.4 shows a nested resampling method that uses in-sample tuning algorithm.

In Figure 6.3, the outer training set is further split into inner training set and inner test set in which the parameter tuning takes place. For each outer training set, the optimal model is obtained. Therefore, from Figure 6.3 and Figure 6.4, there will be three optimal model. The optimal model is then evaluated by using them to predict the value of the target variable for its respective fold. Each fold will output an out-of-sample empirical loss which is average over the three folds.



The inner training set is fitted to the learning function, f, for each parameter. The models is evaluated using the inner test set. Model 1 (with the optimal parameter) is selected.

Model 1 is fitted on the outer training set again and used to predict the value of target variable of the outer test set. The out-of-sample empirical loss is computed (using the first fold as the test set) .

Figure 6.3: Figure of tuning and evaluating stage for out-of-sample tuning method.

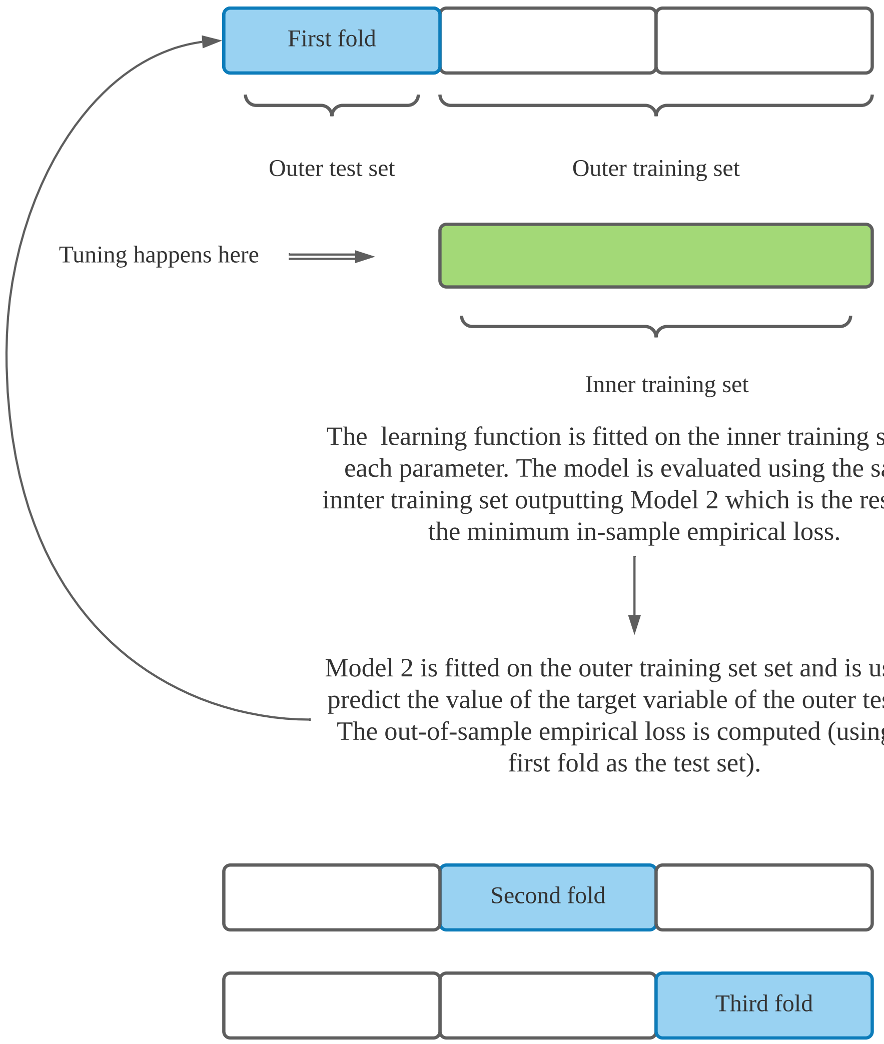

First fold
Outer test set
Outer training set
Tuning happens here
Inner training set
The learning function is fitted on the inner training set for each parameter. The model is evaluated using the same innter training set outputting Model 2 which is the result of the minimum in-sample empirical loss.
Model 2 is fitted on the outer training set set and is used to predict the value of the target variable of the outer test set. The out-of-sample empirical loss is computed (using the first fold as the test set).
Second fold
Third fold

Figure 6.4: Figure to show the process for the tuning and evaluating stage for in-sample tuning method.

6.2.3 Evaluation & Tuning for Distribution Estimation

Here we discuss the evaluation and tuning for unconditional distribution estimation. In this section, we first explain the setting and important terms. Then, we describe the algorithms for evaluation and parameter tuning for distribution estimator.

6.2.3.1 In-sample vs Out-of-sample Evaluation for Distribution Estimation

In this section, we consider a setting for distribution estimation. Let $\mathcal{D} = (Y_1, \ldots, Y_N) \overset{i.i.d}{\sim} Y$ be a training set and $\mathcal{D}^* = (Y_1^*, \ldots, Y_M^*) \overset{i.i.d}{\sim} Y$ be a test set where Y t.v.i $\mathbb{R}$. Let $\hat{f} : \mathbb{R} \to [\mathbb{R} \to \mathbb{R}^+]$ be a PDF estimator, $\hat{F} : \mathbb{R} \to [\mathbb{R} \to [0,1]]$ be the CDF estimator and $\mathcal{L} : \mathbb{R} \times [\mathbb{R} \to \mathbb{R}^+] \to \mathbb{R}$ be the probabilistic loss function for PDF (for CDF $\mathcal{L} : \mathbb{R} \times [\mathbb{R} \to [0,1]] \to \mathbb{R}$). The task is to estimate the distribution defining function at a point. We should point out some important things on distribution estimation:

i. PDF estimation is estimating the PDF at a point (similar to CDF).

ii. The training set $\mathcal{D}$ is used as the sample and the test set $\mathcal{D}^*$ is used as the points where distribution is estimate at.

iii. The estimator PDF, $\hat{f}$, and CDF, $\hat{F}$, estimate (predict) the distribution at a points of the test data $\mathcal{D}^*$.

iv. **In-sample empirical loss:** Evaluates the estimated PDF, $\hat{f}$, and CDF, $\hat{F}$ on training set $\mathcal{D}$ whilst using the same dataset $\mathcal{D}$ for fitting. The in-sample empirical loss is $\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{emp}[\mathcal{L}(\hat{f}, Y_i)]$.

v. **Out-of-sample empirical loss:** Evaluates the estimated PDF, $\hat{f}$, and CDF, $\hat{F}$ on a test set $\mathcal{D}^*$. The out-of-sample empirical loss is $\frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{emp}[\mathcal{L}(\hat{f}, Y_j^*)]$.

The above points are important. However, (iv) and (v) might be confusing. We show by example of what that means. Suppose $\hat{f}$ is a kernel PDF estimator as in Eqn (2.3.4), then using the test set $\mathcal{D}^*$ as the points we want to estimate the distribution using $\mathcal{D}$ as the training set is

$$\hat{f}(Y_j^*) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{Y_j^* - Y_i}{h}\right)$$

where $j = 1, \ldots, M$, K is a kernel function and h is the bandwidth parameter. Evaluating this using the probabilistic empirical loss function will output the out-of-sample empirical loss. However, using the training set $\mathcal{D}$ to predict

$$\hat{f}(Y_j) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{Y_j - Y_i}{h}\right)$$

where $j = 1, \ldots, N$ will lead to in-sample empirical loss.

Evaluating the distribution estimator will follow the same method as in Algorithm 13 and 14, for in-sample and out-of-sample, respectively, but with the absence of the target variable. We show the algorithms for evaluating the empirical loss for distribution estimators in Algorithm 16 and 16.

Algorithm 16 In-sample empirical loss PDF

```
Inputs:
    Inner training data: $\mathcal{D} = (Y_1, ..., Y_N)$; PDF function,
    $\hat{f} : \mathbb{R} \rightarrow [\mathbb{R} \rightarrow \mathbb{R}^+]$; Probabilistic loss function,
    $\mathcal{L} : \mathbb{R} \times [\mathbb{R} \rightarrow \mathbb{R}] \rightarrow \mathbb{R}$.
Output: The in-sample empirical loss, $\mathbb{E}[\mathcal{L}(\hat{f}, Y)]_{emp}$
Steps:
Fit the $\hat{f}$ on $\mathcal{D}$ using the same $\mathcal{D}$ as the observations
Compute the in-sample empirical loss $\mathbb{E}[\mathcal{L}(\hat{f}, Y)]_{emp} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\hat{f}, Y_i)$.
```

Algorithm 17 Out-of-sample empirical loss PDF

```
Inputs:
    Inner training data: $\mathcal{D} = (Y_1, ..., Y_N)$; Inner test data:
    $\mathcal{D}^* = (Y_1^*, ..., Y_M^*)$; PDF function, $\hat{f} : \mathbb{R} \rightarrow [\mathbb{R} \rightarrow \mathbb{R}^+]$;
    Probabilistic loss function, $\mathcal{L} : \mathbb{R} \times [\mathbb{R} \rightarrow \mathbb{R}] \rightarrow \mathbb{R}$
Output: The out-of-sample empirical loss $\mathbb{E}[\mathcal{L}(\hat{f}, Y)]_{emp}$
Steps:
Fit the $\hat{f}$ on $\mathcal{D}^*$ using $\mathcal{D}$ as the observations
Compute the out-of-sample empirical loss $\mathbb{E}[\mathcal{L}(\hat{f}, Y)]_{emp} = \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}(\hat{f}, Y_j^*)$.
```

6.2.3.2 In-sample & Out-of-sample Tuning for Distribution Estimation

Once we have understand how to evaluate for distribution estimation, we can incorporate the evaluation algorithm into tuning to select the parameter. Similar to the regression setting, when using in-sample tuning, increasing the model complexity tends to make the distribution more flexible whereas decreasing the model complexity makes the model less flexible. For example, increasing the value of the bandwidth for a kernel distribution estimator decreases the complexity and will smooth out the shape of the distribution and remove any important features of the distribution. Therefore, using in-sample for tuning causes over-optimism of the in-sample loss (see [2]). We show the algorithm for out-of-sample tuning in Algorithm 18 below.

Algorithm 18 Out-of-sample tuning for PDF estimation

```
Inputs:
    Inner training data: $\mathcal{D} = (Y_1, ..., Y_N)$; Inner test data:
    $\mathcal{D}^* = (Y_1^*, ..., Y_M^*)$; PDF function, $\hat{f} : \mathbb{R} \rightarrow [\mathbb{R} \rightarrow \mathbb{R}^+]$;
    Parameter, $\boldsymbol{p} = p_1, \ldots, p_K$; Loss function, $\mathcal{L} : \mathcal{P} \times \mathbb{R} \rightarrow \mathbb{R}$
Output: A model of $\hat{f}_{p_k}$ with the minimum out-of-sample empirical loss, $\bar{h}_k$
Steps:
for $k = 1, \ldots, K$ do
    Estimate the PDF on $\mathcal{D}^*$ using $p_k$ on the sample $\mathcal{D}$, $\hat{f}_{p_k}(\mathcal{D}^* | \mathcal{D})$
end for
Compute (evaluate) the out-of-sample empirical loss for each bandwidth $p_k$, $\bar{h}_k$
= $\frac{1}{M} \sum_{i=1}^{M} \mathcal{L}(\hat{f}_{p_k}, Y_j^*)$
Select the model $\hat{f}_{p_k}$ with the minimum $\bar{h}_k$ as the tuned model.
```

6.2.3.3 Tuning & Evaluation for Distribution Estimation

Tuning and evaluation for distribution estimation is similar as in Section 6.2.2.1, i.e. tuning takes place in the training stage using the training data and evaluation is done using the test data. The algorithm for tuning and evaluation for distribution estimation is shown in Algorithm 19.

Algorithm 19 Out-of-sample tuning and out-of-sample evaluation for PDF estimation

```
Inputs:
    Training data: $\mathcal{D} = (Y_1, ..., Y_N)$; Test data: $\mathcal{D}^* = Y_1^*, \ldots, Y_M^*$;
    PDF function, $\hat{f} : \mathbb{R} \to [\mathbb{R} \to \mathbb{R}^+]$; Parameter, $\boldsymbol{p} = p_1, \ldots, p_K$;
    Loss function, $\mathcal{L} : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$;
Output: Average out-of-sample empirical loss, $\bar{h}$
Steps:
Split $\mathcal{D}$ into $\mathcal{T} : (Y_1, \ldots, n)$ and $\mathcal{T}^* : (Y_{n+1}^*, \ldots, Y_N^*)$ (where the former is inner training data and the latter is the validation data)
for $p = 1, \ldots, K$ do
    Estimate the PDF of $\mathcal{T}^*$ using $\hat{f}$ on the $\mathcal{T}$ on each $p_k$, $\hat{f}_{p_k}(\mathcal{T}^*|\mathcal{T})$
    Compute the out-of-sample empirical loss for each bandwidth $p_k$, i.e. $h_k = \frac{1}{N-n}\sum_{j=1}^{N-(n)} \mathcal{L}(\hat{f}_{p_k}, Y_j^*)$
    Select the model $\hat{f}_{p_k}$ with the bandwidth with the minimum out-of-sample empirical loss, $p^* = \underset{p}{\operatorname{argmin}}\, h_k$
end for
Estimate the PDF of $\mathcal{D}$ using $p^*$ on the sample $\mathcal{D}^*$, $\hat{f}_{p^*}(\mathcal{D}^*|\mathcal{D})$
Evaluate the out-of-sample empirical loss of $\bar{h} = \frac{1}{M}\sum_{i=1}^{N} \mathcal{L}(\hat{f}_{p^*}, Y_i^*)$
```

6.3 Investigation of In-sample & Out-of-sample Tuning for Distribution Estimation via Log-loss

In this section, we compare the behaviour between in-sample and out-of-sample empirical log-loss for tuning the bandwidth using Gaussian kernel PDF. Here, we aim to tune the bandwidth of kernel PDF estimator using in-sample and out-of-sample empirical loss. We provide theoretical proof to investigate the tuning algorithm for distribution estimation. Firstly, we explain in details about the two cases we are investigating. Then, we define the settings which include the datasets used for in-sample and out-of-sample tuning. Then, we specify the estimator which is the univariate Gaussian kernel PDF and define the empirical log-loss for the estimator. To assist the proof, we present some preliminaries definitions and lemmas.

In the proof, we compare the two settings: (1) test set is a subset of the training set during tuning; (2) test set is **not** a subset of training set during tuning. We show that for (1), this result to the in-sample empirical loss which is unbounded when it is a function of the bandwidth. The in-sample empirical loss decreases to infinity as the value of bandwidth decreases to 0. However, we show under (2), the out-of-sample empirical loss is bounded and we further show that there exist a global minimum (which signifies the optimal bandwidth). Under this condition, it signifies that even when the test set contain one new point not present in the training set, the proof still holds. This concludes that the present of one new data point in the test set will allow the out-of-sample empirical loss to achieve a minimum point.

6.3.1 Theoretical Proof: Tuning Kernel PDF via Log-loss

For this investigation, we proposed two cases to differentiate the in-sample tuning and out-of-sample tuning.

1. **Test set is a subset of training set:** There are two subcases that fall under this: (1) test set is equal to the training set; (2) test set is a subset of the training set (i.e. all data points of the test set are in the training set but not all data points in the training set are in the test set). This is use to evaluate the in-sample empirical log-loss. For this case, we hypothesis that minimum does not exist. The in-sample empirical loss will decrease as the bandwidth tends to 0 and increases as the bandwidth tends to infinity.
2. **Test set is not a subset of the training set:** There are two cases that fall into this: (1) test set is not equal to the training set (i.e. there is no over lapping data points in both sets); (2) test set contain one data point not in the training set. This is for the out-of-sample empirical log-loss. Under this case, we hypothesise there exist a minimum point and that the out-of-sample empirical loss is bounded.

To investigate the impact of the two cases above on tuning the bandwidth, we inspect the behaviour of the empirical loss as the bandwidth tends to 0 and as the bandwidth tends to ∞. Therefore, we show the proof of the limiting behaviour of the in-sample and out-of-sample empirical loss as $h \to 0$ and as $h \to \infty$ for both cases. The set up of the proof is in Section 6.3.1.1, the preliminary definitions and lemmas in Section C.1 and Section 6.3.1.3, respectively. The main proof is in Section 6.3.2.

6.3.1.1 Setting

Here, we present setting, estimators and the definitions and lemmas to support the proof.

Cases

Let $\mathcal{D}$ be a vector of the training data, $\mathcal{D} := (Y_1, ..., Y_N)^T \in \mathbb{R}^N$ and let $\mathcal{D}^*$ be a vector of the test data, $\mathcal{D}^* := (Y_1^*, ..., Y_M^*)^T \in \mathbb{R}^M$. The cases are:

1. Let $M, N \in \mathbb{N}$. Suppose $\mathcal{D}^* \subseteq \mathcal{D}$, then we have $M \leq N$. Without loss of generality, for all $i \in (1, 2, ..., M)$ and $j \in (1, 2, ..., M)$, we have $Y_i - Y_j^* = 0$ when $j = i$.
2. Let $M, N \in \mathbb{N}$ and let $\mathcal{D}^* \nsubseteq \mathcal{D}$.

6.3.1.2 List of Definition

Definition 6.3.1. ***Empirical log-loss function using Gaussian kernel***
Let $\mathcal{D}$ be a vector of the training data, $\mathcal{D} := (Y_1, ..., Y_N)^T \in \mathbb{R}^N$ and let $\mathcal{D}^*$ be a vector of the test data, $\mathcal{D}^* := (Y_1^*, ..., Y_M^*)^T \in \mathbb{R}^M$. Then, let $f : \mathbb{R}^+ \to \mathbb{R}$ be a function of h be defined by

$$f(h) = \frac{1}{M} \sum_{j=1}^{M} \left(-\log \left[\frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} \exp \left\{ -\frac{1}{2} \left(\frac{t_{ij}}{h} \right)^2 \right\} \right] \right) \tag{6.3.1}$$

where $t_{ij} = Y_i - Y_j^*$ and Y_i, Y_j^* are the training data of size N and test set of size M respectively. For simplicity, we let u_{ij} be

$$u_{ij} = \exp \left\{ -\frac{1}{2} \left(\frac{t_{ij}}{h} \right)^2 \right\} \tag{6.3.2}$$

in eqn (6.3.1) can be expressed as

$$f(h) = \frac{1}{M} \sum_{j=1}^{M} \log(Nh\sqrt{2\pi}) - \frac{1}{M} \sum_{j=1}^{M} \log \left(\sum_{i=1}^{N} u_{ij} \right) \tag{6.3.3}$$

$$= \log(Nh\sqrt{2\pi}) - \frac{1}{M} \sum_{j=1}^{M} \log \left(\sum_{i=1}^{N} u_{ij} \right) \tag{6.3.4}$$

6.3.1.3 List of Lemmas

Lemma 6.3.1. *Let f, $\mathcal{D}$ and $\mathcal{D}^*$ be defined as in Def 6.3.1. f is continuous when $h > 0$.*

Proof. From the definition of continuous function, a logarithmic function is always continuous on its domain. An exponential function is always continuous in the domain $\mathbb{R}$. Then, by the property of a continuous function, a composition of continuous functions is also continuous. Hence, a composition of logarithmic functions with the domain greater than 0 is continuous which is also true for a composition of exponential functions. Hence, by definition of a continuous function and the properties of logarithmic and exponential functions, f is always continuous when $h > 0$. □

Lemma 6.3.2. *Let $g : \mathbb{R}^+ \to \mathbb{R}$ be defined as $g(h) = \log(Nh\sqrt{(2\pi)})$. Then, under Def C.1.1, g is not bounded below such that*

$$\lim_{h\to 0} g(h) = \infty.$$

Proof. Under Def C.1.1), $g(h)$ is not bounded below when $h \to 0$ if every $L < 0$ there is $h_1 > 0$ such that $\log(Nh\sqrt{2\pi}) < L$ whenever $0 < |h - 0| < h_1$. Then, we need to find h_1 such that $\log(h\sqrt{2\pi}) < L$.

$$\log(Nh\sqrt{2\pi}) < L \Leftrightarrow Nh\sqrt{2\pi} < \exp(L) \Leftrightarrow h < \frac{\exp(L)}{N\sqrt{2\pi}}.$$

Choosing $h_1 = \frac{\exp(L)}{N\sqrt{2\pi}}$. Since $0 < |h - 0| < h_1$, we have $0 < |h| < \frac{\exp(L)}{N\sqrt{2\pi}}$. Hence, $0 < |Nh\sqrt{2\pi}| < \exp\{L\}$ and by taking the log, we will get $-\infty < \log(Nh\sqrt{2\pi}) \leq L$. □

Lemma 6.3.3. *Let $g : \mathbb{R}^+ \to \mathbb{R}$ be defined as $g(h) = \log(Nh\sqrt{(2\pi)})$. Then, under Def C.1.1, $g(h)$ tends to ∞ such that*

$$\lim_{h\to \infty} g(h) = \infty.$$

Proof. Under Def C.1.1), $g(h)$ goes to ∞ as $h \to \infty$ if for every $U > 0$ there is $\delta > 0$ such that $\log(h\sqrt{2\pi}) > U$ whenever $h > \delta$. Then, we need to find δ such that $\log(h\sqrt{2\pi}) > U$.

$$\log(h\sqrt{2\pi}) > U \Leftrightarrow Nh\sqrt{2\pi} > \exp(U) \Leftrightarrow h > \frac{\exp(U)}{N\sqrt{2\pi}}.$$

Choosing $h_2 = \frac{\exp(U)}{N\sqrt{2\pi}}$, we obtain $h > \frac{\exp(U)}{N\sqrt{2\pi}}$ since $h > \delta$. Hence, we can show that $\log(h\sqrt{2\pi}) > U$. □

Lemma 6.3.4. *Let $M, N \in \mathbb{N}$. Then, for $\mathcal{D}^* \nsubseteq \mathcal{D}$, we have a matrix*

$$T = (t_{ij})_{i,j=1}^{N,M}$$

such that $t_{ij} = Y_i - Y_j^*$ in which a column of matrix T with a non-zero entry.

Proof. Let $\mathcal{D}^* \nsubseteq \mathcal{D}$, then we have

$$\begin{aligned}
&\Rightarrow \exists Y_j^* \in \mathcal{D}^* : Y_j^* \notin \mathcal{D} \\
&\Rightarrow \exists Y_j^* \in \mathcal{D}^* \, \forall Y_i \in \mathcal{D} : Y_i - Y_j^* \neq 0 \\
&\Rightarrow \exists j \in \{1, ..., M\} \forall i \in \{1, ..., N\} : t_{ij} \neq 0
\end{aligned}$$

□

Lemma 6.3.5. *Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function. Let $K \subset \mathbb{R}$ be a compact set. Then, $f(K) = \{f(x) : x \in K\}$ is a compact set.*

Proof. Refer to [110]. □

6.3.2 Main Proofs: Proofs of Boundaries

In this section, we provide the proof of boundaries of the empirical log-loss of Gaussian kernel PDF for the two cases: (1) test set is a subset of the training set; (2) test set is not a subset of the training set. For both cases, we show the limit boundary of the empirical log-loss of Gaussian kernel PDF as the bandwidth tends to 0 and infinity.

6.3.2.1 Proposition 1

Below is the proof of the proposition to show that the log-loss is unbounded from below if the test set is a subset of training set.

Proposition 6.3.1. *Let $f : \mathbb{R}^+ \to \mathbb{R}$ be as defined in Def 6.3.1. Under Case 1 Section 6.3.1.1, f is not bounded from below which is implied by the fact that*

$$\lim_{h \to 0} f(h) = -\infty$$

Proof. Suppose that f as defined in Def 6.3.1. Under Case 1, we can rewrite f as in (6.3.4),

$$f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M} \sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \tag{6.3.5}$$

where u_{ij} as defined in Eqn (6.3.2). From Case 1 (Section 6.3.1.1), $u_{ii} = 1 \forall i, j \in \{1, ..., M\}$ and $0 < u_{ij} \leq 1$ if $i \neq j$ where $i \in \{1, ..., N\}$. Then,

$$\sum_{i=1}^{N} u_{ij} = u_{ii} + \sum_{i \neq j}^{N} u_{ij} = 1 + \sum_{i \neq j}^{N} u_{ij} \geq 1 \qquad \text{for any } j \in \{1, ..., M\} \tag{6.3.6}$$

We take $\log$ on both sides of (6.3.6) to obtain

$$\log \sum_{i=1}^{N} u_{ij} \geq \log(1) = 0 \tag{6.3.7}$$

because $\log$ is an increasing function. By averaging over j and multiplying both sides by -1, we deduce

$$-\frac{1}{M} \sum_{j=1}^{M} \log \sum_{i=1}^{N} u_{ij} \leq 0. \tag{6.3.8}$$

Finally, adding $\log(Nh\sqrt{2\pi})$ to eqn (6.3.8),

$$\begin{aligned} f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M} \sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) &\leq \log(Nh\sqrt{2\pi}) - \frac{1}{M}(0) \\ &= \log(Nh\sqrt{2\pi}). \end{aligned} \tag{6.3.9}$$

Under Lemma 6.3.2, we found that $\lim_{h \to 0} \log(Nh\sqrt{2\pi}) = -\infty$. From eqn (6.3.9), since $f(h) < log(Nh\sqrt{(2\pi)})$, we can conclude that $\lim_{h \to 0} f(h) = -\infty$ Thus, under Case 1, $f(h)$ is unbounded from below when $h \to 0$. □

6.3.2.2 Proposition 2

Below is the proof of the proposition to show that the log-loss is unbounded from above if the test set is a subset of training set.

Proposition 6.3.2. *Let $f : \mathbb{R}^{+} \to \mathbb{R}$ be as defined in Def 6.3.1. Under Case 1 Section 6.3.1.1, f tends to ∞ as $h \to \infty$ i.e.*

$$\lim_{h \to \infty} f(h) = \infty.$$

Proof. Suppose that f is defined as in Eqn (6.3.1). Under Case 1 in Section 6.3.1.1, we rewrite f as

$$f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \tag{6.3.10}$$

where u_{ij} as defined in Eqn (6.3.2). From Case 1, $u_{ii} = 1 \forall i, j \in \{1, ..., M\}$ and $0 < u_{ij} \leq 1$ if $i \neq j$ where $i \in \{1, ..., N\}$. Then,

$$\sum_{i=1}^{N} u_{ij} \leq N \qquad \text{for any } j \in \{1, ..., M\} \tag{6.3.11}$$

We take $\log$ on both sides to obtain

$$\log \sum_{i=1}^{N} u_{ij} \leq \log(N) \qquad \text{for any } j \in \{1, ..., M\} \tag{6.3.12}$$

because $\log$ is an increasing function. By multiplying both sides by -1 and averaging over j, we deduce

$$-\frac{1}{M}\sum_{j=1}^{M} \log \sum_{i=1}^{N} u_{ij} \geq -\frac{1}{M}\sum_{j=1}^{M} \log(N) = -\log(N). \tag{6.3.13}$$

Finally, adding $\log(Nh\sqrt{2\pi})$ we obtain

$$f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \geq \log(Nh\sqrt{2\pi}) - \log(N) = \log(h\sqrt{2\pi}). \tag{6.3.14}$$

By Lemma (6.3.3), we found that $\lim_{h\to\infty} \log(h\sqrt{2\pi}) = \infty$. Since $f(h) \geq \log(h\sqrt{2\pi})$, we can conclude that $f(h)$ tends to ∞ as $h \to \infty$. □

6.3.2.3 Lemma 1

In this section, we will provide proof of the lemma (Lemma 6.3.6) for the log-loss of univariate Gaussian kernel estimator when the bandwidth goes to 0 under Case 2 in Section 6.3.1.1 (i.e. the test set is not a subset of the training set).

Lemma 6.3.6. *Let $f : \mathbb{R}^+ \to \mathbb{R}$ be as defined in Definition 6.3.1. Under Case 2 in Section 6.3.1.1,*

$$\lim_{h\to 0} f(h) = \infty. \tag{6.3.15}$$

Proof. For the case $\mathcal{D}^* \nsubseteq \mathcal{D}$, we refer by Lemma 6.3.4. Then, under Case 2 in Section 6.3.1.1, let $Q = (u_{ij})_{i,j=1}^{N,M}$ where u_{ij} as defined in Eqn (6.3.2). Let $p \in \mathbb{N} \cup \{0\}$ be the number of columns of Q that has entries '1' and $J \subseteq \{1, 2, ..., M\}$ be the set that contain the p columns while $\tilde{J} = \{1, 2, ..., M\} \setminus J$.

$$\begin{aligned} \forall j \in J : \qquad & \sum_{i=1}^{N} u_{ij} \leq N \\ \Rightarrow \forall j \in J : \qquad & \log\left(\sum_{i=1}^{N} u_{ij}\right) \leq \log(N) \\ \Rightarrow \sum_{j \in J} & \log\left(\sum_{i=1}^{N} u_{ij}\right) \leq p\log(N). \end{aligned} \tag{6.3.16}$$

For $(M-p)$ columns that are without entries 1, i.e. for $j \in \tilde{J}$, we have:

$$\begin{aligned} \sum_{i=1}^{N} u_{ij} &\leq N v_j \quad \text{where } v_j \text{ is the largest } u_{ij} \text{ in column } j \in \tilde{J} \\ \log\left(\sum_{i=1}^{N} u_{ij}\right) &\leq \log(N) + \log(v_j) \\ \sum_{j \in \tilde{J}} \log\left(\sum_{i=1}^{N} u_{ij}\right) &\leq (M-p)\log(N) + \sum_{j \in \tilde{J}} \log(v_j) \end{aligned} \tag{6.3.17}$$

Then, add eqn (6.3.16) and eqn (6.3.17) together to obtain

$$\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \leq M\log(N) - \sum_{j \in \tilde{J}}^{M} \frac{(t_j)^2}{2h^2} \text{ where } t_j \text{ corresponds to the largest } v_j \text{ for column } j \in \tilde{J} \tag{6.3.18}$$

Multiplying eqn (6.3.18) by $-\frac{1}{M}$ we obtain,

$$-\frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \geq -\frac{1}{M}\left[M\log(N) - \sum_{j \in \tilde{J}}^{M} \frac{(t_j)^2}{2h^2}\right]. \tag{6.3.19}$$

Finally, we add $\log(Nh\sqrt{2\pi})$ to obtain

$$\begin{aligned} f(h) &= \log(Nh\sqrt{2\pi}) - \frac{1}{M}\sum_{j=1}^{M}\log\left(\sum_{i=1}^{N}u_{ij}\right) \\ &\geq \log(Nh\sqrt{2\pi}) - \log(N) - \sum_{j\in\tilde{J}}^{M}\frac{(t_j)^2}{2Mh^2} \\ &\geq \log(h\sqrt{2\pi}) + \frac{\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2Mh^2}. \end{aligned} \tag{6.3.20}$$

Reparameterize $\gamma = \frac{1}{h}$. We can rewrite eqn (6.3.20) in terms of γ

$$f(\gamma^{-1}) \geq \log(\gamma^{-1}\sqrt{2\pi}) + \frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M}$$

Then, we can show that

$$\begin{aligned} \lim_{\gamma\to\infty} f(\gamma^{-1}) &\geq \lim_{\gamma\to\infty}\left(\log(\gamma^{-1}\sqrt{2\pi}) + \frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M}\right) \\ \lim_{\gamma\to\infty} f(\gamma^{-1}) &\geq \lim_{\gamma\to\infty}\left(\log(\sqrt{2\pi}) - \log(\gamma) + \frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M}\right). \end{aligned} \tag{6.3.21}$$

On the RHS of eqn (6.3.21), $-log(\gamma) \to -\infty$ while $\frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M} \to \infty$ as $\gamma \to \infty$. To apply L'Hôpital's rule, we first take the exponential of eqn (6.3.21) to obtain

$$\lim_{\gamma\to\infty} \exp\left\{f(\gamma^{-1})\right\} \geq \lim_{\gamma\to\infty}\left[\exp(A) \times \frac{\exp\left\{\frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M}\right\}}{\gamma}\right] \tag{6.3.22}$$

where $A = \sqrt{2\pi}$. Applying L'Hôpital's rule from Theorem 2.4.1 [110], where the numerator and denominator are both differentiated w.r.t γ, we obtain

$$\lim_{\gamma\to\infty} \exp\left\{f(\gamma^{-1})\right\} \geq \lim_{\gamma\to\infty}\left[\exp(A) \times \frac{\frac{\gamma\sum_{j\in\tilde{J}}^{M}(t_j)^2}{M}\exp\left\{\frac{\gamma^2\sum_{j\in\tilde{J}}^{M}(t_j)^2}{2M}\right\}}{1}\right] = \infty. \tag{6.3.23}$$

Since the numerator of the above goes to ∞ as $\gamma \to \infty$ (since $\sum_{j\in\tilde{J}} t_j^2 > 0$), then

$\lim_{\gamma\to\infty} \exp f(\gamma^{-1}) = \infty$. By Definition C.1.2, this means we have

$$\forall c_2 > 0 \; \exists \gamma_2 > 0 : \gamma > \gamma_2 \Rightarrow \exp\{f(\gamma^{-1})\} > c_2. \tag{6.3.24}$$

To show that $\lim_{\gamma\to\infty} f(\gamma^{-1}) = \infty$, by Definition C.1.2,

$$\forall c_1 > 0 \; \exists \gamma_1 > 0 : \gamma > \gamma_1 \Rightarrow f(\gamma^{-1}) > c_1. \tag{6.3.25}$$

Let $c_1 > 0$ be given. Then, we have

$$\forall c_2 > 0 \exists \gamma_1 > 0 \forall \gamma > \gamma_1 : \exp(f(\gamma^{-1})) > c_2. \tag{6.3.26}$$

Choosing $c_2 = \exp(c_1)$, eqn (6.3.26) implies

$$\begin{aligned} &\exists \gamma_1 > 0 \; \forall \gamma > \gamma_1 : \exp(f(\gamma^{-1})) > \exp\{c_1\} \\ \Leftrightarrow &\exists \gamma_1 > 0 \; \forall \gamma > \gamma_1 : f(\gamma^{-1}) > c_1 \end{aligned} \tag{6.3.27}$$

Hence, $\lim_{\gamma\to\infty} f(\gamma^{-1}) = \infty$ which implies $\lim_{h\to 0} f(h) = \infty$. □

6.3.2.4 Lemma 2

In this section, we will provide a proof of the log-loss of a univariate Gaussian kernel tends to ∞ when the bandwidth goes to ∞ under Case 2 in Section 6.3.1.1 i.e. test set is not a subset of the training set.

Lemma 6.3.7. *Let $f : \mathbb{R}^+ \to \mathbb{R}$ be as defined in Def 6.3.1 and let $h \in \mathbb{R}^+$. Under Case 2 in Section 6.3.1.1, f tends to ∞, as $h \to \infty$ i.e.*

$$\lim_{h\to\infty} f(h) = \infty.$$

Proof. Suppose that f is defined as in Def 6.3.1. Under Case 2 in Section 6.3.1.1, we can rewrite f as

$$f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \tag{6.3.28}$$

Under Case 2,

$$\begin{aligned} \sum_{i=1}^{N} u_{ij} &\leq N \qquad \text{for any } j \in \{1, ..., M\} \\ \Rightarrow \log\left(\sum_{i=1}^{N} u_{ij}\right) &\leq \log(N) \qquad \text{for any } j \in \{1, ..., M\} \\ \Rightarrow \sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) &\leq \sum_{j=1}^{M} \log(N) = M \log(N). \end{aligned}$$

Then, dividing the above equation by M and multiplying by -1,

$$-\frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \geq -\log(N). \tag{6.3.29}$$

and lastly adding $\log(Nh\sqrt{2\pi})$, we obtain

$$f(h) = \log(Nh\sqrt{2\pi}) - \frac{1}{M}\sum_{j=1}^{M} \log\left(\sum_{i=1}^{N} u_{ij}\right) \geq \log(Nh\sqrt{2\pi}) - \log(N) = \log(h\sqrt{2\pi}) \tag{6.3.30}$$

By Lemma 6.3.3, we found that $\lim_{h\to\infty} \log(h\sqrt{2\pi}) = \infty$. Since $f(h) > \log(Nh\sqrt{2\pi})$ by Eqn (6.3.30), we can conclude that under Case 2, $f(h)$ tends to ∞ as $h \to \infty$. □

Lemma 6.3.6 and 6.3.7 are for general case when we have overlapping data points in $\mathcal{D}$ and $\mathcal{D}^*$. However, the lemmas also work for when $M = 1$ such that $\mathcal{D}^* \nsubseteq \mathcal{D}$ and $\mathcal{D}^* \neq \mathcal{D}$. This shows that the lemmas only need at least one new point in $\mathcal{D}^*$ that is not present in the $\mathcal{D}$.

6.3.2.5 Lemma 3

In this section, we will provide a proof to show that in the case where $\mathcal{D}^* \nsubseteq \mathcal{D}$, there exist a minimum of the out-of-sample empirical log-loss.

Lemma 6.3.8. *Let $f : \mathbb{R}^+ \to \mathbb{R}$ be the function defined in Def 6.3.1. Let $\mathcal{D}$ be a vector of training data, $\mathcal{D} : (Y_1, ..., Y_N)^T \in \mathbb{R}^N$ and let $\mathcal{D}^*$ be a vector of test data $\mathcal{D}^* = (Y_1^*, ..., Y_M^*)^T \in \mathbb{R}^M$. Let $\mathcal{D}^* \nsubseteq \mathcal{D}$. Then,*

1. $\lim_{h\to 0} f(h) = \infty$
2. $\lim_{h\to \infty} f(h) = \infty$
3. *There exist $L \in \mathbb{R}$ such that $\forall h > 0 : f(h) \geq L$*

Proof. 1. See Lemma 6.3.6

2. See Lemma 6.3.7

3. Let $c_1 \in \mathbb{R}^+$. Suppose there is $h_1 > 0$, then $f(h_1) > 0$. From Lemma 6.3.6, there is $h_2 > 0$ for all $h < h_2$ such that $f(h) > c_1$. Following Lemma 6.3.7, there is $h_3 > 0$ for all $h > h_3$ such that $f(h) > c_1$. Let $\tilde{h}_2 = \min(h_2, h_3)$ and $\tilde{h}_3 = \max(h_2, h_3)$. Also, we let $f(h_1) = c_1$. We know that $f(h) > c_1$ if $h > \tilde{h}_3$ or $h < \tilde{h}_2$. Then we have $f(h) \leq c_1$ when $h \in [\tilde{h}_2, \tilde{h}_3]$ where the compact interval $I = [\tilde{h}_2, \tilde{h}_3]$ will always be none empty because h_1 will always be in I. Then, we can show that $\forall h \notin I : f(h) > c_1$ and $\exists h \in I : f(h) \leq c_1$. By Lemma 6.3.5, for any interval $I = [\tilde{h}_2, \tilde{h}_3]$, $f(I) = \{f(h) : h \in I\}$ is a compact set and we know that there exist a minimum in the compact set $f(I)$. We also know that $f(h) > minf(I) = L$ if $h \in I$. Hence, we can conclude that $f(h) \geq \min(0, \min f(I))$ for any $h > 0$.

□

6.3.2.6 Theorem on Global Minimum of Out-of-sample Empirical Log-loss of Gaussian Kernel pdf

In this final theorem below, we will provide the a proof in the case where $\mathcal{D}^* \subsetneq \mathcal{D}$, there exist a minimum and that minimum point is the global minimum for out-of-sample empirical log-loss.

Theorem 6.3.1. *Let f, $\mathcal{D}$ and $\mathcal{D}^*$ be defined in Definition 6.3.1. Then, the followings are equivalent*

1. $\mathcal{D}^* \nsubseteq \mathcal{D}$
2. *f has a global minimum .*

Proof. Here, we present the proof in two directions:

1. (1) $\rightarrow$ (2).
   If $\mathcal{D}^* \nsubseteq \mathcal{D}$, then f has a global minimum
2. (2) $\rightarrow$ (1) by using contraposition, $\neg(1) \rightarrow \neg(2)$
   If $\mathcal{D}^* \subseteq \mathcal{D}$ then f has no global minimum.

We construct the first proof. By Lemma 6.3.1, f is a continuous function. Suppose there is $h_1 > 0$, then $f(h_1) > 0$. From Lemma 6.3.6, there is $h_2 > 0$ for all $h < h_2$ such that $f(h) > c_1$. Following Lemma 6.3.7, there is $h_3 > 0$ for all $h > h_3$ such that $f(h) > c_1$. Let $\tilde{h}_2 = \min(h_2, h_3)$ and $\tilde{h}_3 = \max(h_2, h_3)$. Also, we let $f(h_1) = c_1$. We know that $f(h) > c_1$ if $h > \tilde{h}_3$ or $h < \tilde{h}_2$. Then we have $f(h) \leq c_1$ when $h \in [\tilde{h}_2, \tilde{h}_3]$ where the compact interval $I = [\tilde{h}_2, \tilde{h}_3]$ will always be none empty because h_1 will always be in I. Then, we can show that $\forall h \notin I : f(h) > c_1$ and $\exists h \in I : f(h) \leq c_1$. By Lemma 6.3.5, for any interval $I = [\tilde{h}_2, \tilde{h}_3]$, $f(I) = \{f(h) : h \in I\}$ is a compact set. Hence, by extreme value theorem, there exist a minimum when f is continuous on the compact set I. Since $\forall h \notin I : f(h) > c_1$ we know that there is no h outside of I will give $f(h) < c_1$, hence the minimum in the compact set I is also the global minimum.

For the second proof, we prove this by contraposition, such that if $\mathcal{D}^* \subseteq \mathcal{D}$, then f has no global minimum. This is easily seen by Proposition 6.3.1 and Proposition 6.3.2, where under assumption 1, $\lim_{h \to 0} f(h) = -\infty$ and $\lim_{h \to \infty} f(h) = \infty$, respectively. Hence, there is no global minimum when $\mathcal{D}^* \subseteq \mathcal{D}$. □

6.4 Investigation of Out-of-sample Tuning for Distribution Estimation via PSL

In this section, we investigate the behaviour of the out-of-sample empirical PSL for tuning the bandwidth of a kernel PDF estimator. This investigation is motivated by [6] and [5] where it stated there exist a threshold for out-of-sample empirical PSL to be bounded as the bandwidth to the kernel PDF goes to 0 and infinity. This situation is due to the discretization of the real world data. In this section, we provide a formal proof of the out-of-sample empirical PSL for Gaussian PDF when the bandwidth goes to 0 and ∞.

In the proof, we compared two settings: (1) The total number of test data points to the number of data points that exist in both training and test sets is less than $2\sqrt{2}$; (2) The total number of test data points to the number of data points that exist in both training and test sets is greater than $2\sqrt{2}$. From (1), the out-of-sample empirical PSL for Gaussian PDF goes to $-\infty$ and 0 as the bandwidth goes to 0 and ∞, respectively. Whereas, (2) proves that the out-of-sample empirical PSL for Gaussian PDF goes to ∞ and to 0 as the bandwidth goes to 0 and ∞, respectively. This concludes that the ratio total number of test data points to the number of data points that exist in both training and test sets must be greater than $2\sqrt{2}$ to achieve a minimum point. We also proved that under this condition a global minimum is achieved.

Firstly, we explain in detail the conjecture from [5] and proposed a new setting to investigate the behaviour of out-of-sample empirical PSL for Gaussian kernel PDF. Then, we define the setting for the proof. Then, we define the function to prove the boundary of out-of-sample empirical PSL. Lastly, we present the formal proof.

6.4.1 Theoretical Proof: Out-of-sample Tuning for Gaussian Kernel PDF via PSL

Silverman's Conjecture

Let $\mathcal{D} = (Y_1, \dots, Y_N) \overset{i.i.d}{\sim} Y$ where Y t.v.i $\mathbb{R}$. Let $\kappa \in \mathbb{N}$ be a number of pairs $i < j$ for which $Y_i = Y_j$ where $i, j = 1, \dots, N$. Then there exist a threshold β ([6]), such that

$$\beta = \frac{K^{(2)}(0)}{2(2K(0) - K^{(2)}(0))}.$$

where K is the kernel function and $K^{(2)}$ is convolution of the kernel function with itself. When $\kappa > \beta N$ (N is the total number of data points in $\mathcal{D}$), the empirical PSL will tend to $-\infty$ as the bandwidth of the Gaussian PDF tends to 0. [5] used the same dataset $\mathcal{D}$ for evaluation and training and shows that the threshold for a Gaussian kernel PDF is $\beta = 0.55$.

New Setting

To ensure that the tuning is via out-of-sample, it is important to separate the datasets. Consider a training set $\mathcal{D} = (Y_1, \dots, Y_N) \overset{i.i.d}{\sim} Y$ where Y t.v.i $\mathbb{R}$. Consider a test set $\mathcal{D}^* = (Y_1^*, \dots, Y_M^*) \overset{i.i.d}{\sim} Y$ where Y t.v.i $\mathbb{R}$. Let $M = m + n$, where m is the number of test points $\mathcal{D}^*$ that does not exist in $\mathcal{D}$ and $n \leq N$ be the number of test points $\mathcal{D}^*$ that overlaps with $\mathcal{D}$. We hypothesise that for a Gaussian kernel PDF,

i. when $M < 2\sqrt{2}n$, we obtain the same result as [5] where the out-of-sample empirical PSL is unbounded (i.e. out-of-sample empirical PSL tends to $-\infty$ as $h \to 0$).

ii. when $M > 2\sqrt{2}n$ the out-of-sample empirical PSL tends to ∞ and 0 as $h \to 0$ and ∞, respectively.

Suppose we have N, n, m as in the new setting. It is found that $\frac{m}{n} < 2\sqrt{2} - 1 = \frac{1}{0.55}$. This shows the relationship between the result of the new setting with [5]'s.

Therefore, we provide a proof for the limit of boundaries of the out-of-sample empirical PSL for Gaussian kernel when $h \to 0$ and $h \to \infty$ for the two cases: (1) $M < 2\sqrt{2}n$: (2)$M > 2\sqrt{2}n$. The purpose is to provide a formal proof for [5]'s statement focussing on Gaussian kernel PDF and to ensure clarity of using out-of-sample method. The set up of the proof is shown in Section 6.4.1.1. The formal proof is shown in Section 6.4.2.

6.4.1.1 Settings for PSL

Here, we describe the set up where we define the training set and the test set, the Gaussian kernel PDF and the out-of-sample empirical PSL. Then, we will list down definitions and lemmas that are important to support our main proof.

Dataset

Let $M, N, n \in \mathbb{N}$. Define the vectors

$$\mathcal{D} = (Y_1, \dots, Y_N) \in \mathbb{R}^N$$

and

$$\mathcal{D}^* = (Y_1^*, \dots, Y_n^*, Y_{n+1}^*, \dots, Y_M^*) \in \mathbb{R}^M.$$

We call $\mathcal{D}$ as the vector of training set and $\mathcal{D}^*$ as the vector of test set and $\mathcal{D}^* \not\subseteq \mathcal{D}$. Define $m = M - n$ (i.e. $M = n + m$). We refer n as number of test points in $\mathcal{D}^*$ that exist in $\mathcal{D}$ and $n \leq N$, i.e. $Y_j^* - Y_i = 0$ for all $j = 1, \dots, n, n+1, \dots, M$ and $i = 1, \dots, n$. Then, m is be the number test data $\mathcal{D}^*$ not $\mathcal{D}$.

List of Definitions

Let the PSL is define in Eqn (3.3.11) and the PDF estimator is defined in Eqn (2.3.4). We define the out-of-sample empirical PSL for univariate Gaussian kernel PDF in Def 6.4.1.

Definition 6.4.1. ***The out-of-sample empirical PSL for univariate Gaussian kernel PDF***

Let $\mathcal{D}$ be a vector of training data, $\mathcal{D} = (Y_1, \dots, Y_N) \in \mathbb{R}^N$ and $\mathcal{D}^*$ let be a vector of test data, $\mathcal{D}^* = (Y_1^*, \dots, Y_M^*) \in \mathbb{R}^M$. We define the function $g : \mathbb{R}^+ \to \mathbb{R}$, as

$$g(h) = -\frac{2}{hNM\sqrt{2\pi}} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)} + \frac{1}{2N^2h\sqrt{\pi}} \sum_{j=1}^{N} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)^2} \tag{6.4.1}$$

where

i. Y_i, Y_j^* are the training data of size N and test data of M, respectively *ii.* $t_{ij} = Y_j^* - Y_i$ *for* $i = 1, \ldots, N$ *and* $j = 1, \ldots, M$

iii. $r_{ij} = Y_j - Y_i$ *for* $i, j = 1, \ldots, N$

iv. $h \in \mathbb{R}^+$ *is the bandwidth.*

Lemma 6.4.1. *Let* g *be defined as in Def 6.4.1.* g *is continuous when* $h > 0$.

Proof. From the definition of continuous function, an exponential function is always continuous in the domain $\mathbb{R}$. An exponential function with the domain greater than 0 is continuous which is also true for a summation of exponential functions. Hence, by definition of a continuous function and the properties of exponential functions, g is always continuous when $h > 0$. □

6.4.2 Main Proofs

In this section, we provide the main proofs for our investigation. We provide the proofs of the limit boundaries of Eqn 6.4.1 as $h \to 0$ and $h \to \infty$ for each cases: (1) $M > 2\sqrt{2}n$; (2) $M \leq 2\sqrt{2}n$.

Lemma 6.4.2. *Define the function* $g : \mathbb{R}^+ \to \mathbb{R}$ *as in Eqn (6.4.1). Then, for* M *and* n *from Section 6.4.1.1,*

1. *when* $M > 2\sqrt{2}n$, g *tends to* ∞ *as* $h \to 0$, *i.e.*

$$\lim_{h \to 0} g(h) = \infty. \tag{6.4.2}$$

2. *when* $M < 2\sqrt{2}n$, g *tends to* $-\infty$ *as* $h \to 0$, *i.e.*

$$\lim_{h \to 0} g(h) = -\infty. \tag{6.4.3}$$

Proof. Recall the Eqn (6.4.1). We re-write the function again,

$$g(h) = -\frac{2}{hNM\sqrt{2\pi}} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)} + \frac{1}{2N^2 h\sqrt{\pi}} \sum_{j=1}^{N} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)^2}. \tag{6.4.4}$$

For simplicity, we make the denominator equal

$$g(h) = \frac{1}{h} \cdot \frac{-2(2N) \sum_{j=1}^{M} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)} + M\sqrt{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)^2}}{2MN^2\sqrt{2\pi}}. \tag{6.4.5}$$

To prove the two cases, we firstly show the limit of function g as $h \to 0$ before we consider the cases. This proof uses a sequence of elementary computation and elementary limit.

Let $u_{ij} = \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)^2}$ and $w_{ij} = \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)^2}$ for $i = 1, \dots, N$ and $j = 1, \dots, M$.

The first term of the numerator of Eqn (6.4.5) depends on the training vector and the test vector. While, the second term of the numerator only depends on the training vector. Consider the first term of the numerator of Eqn (6.4.5). Since we define $M = m + n$, we can express the summation as

$$\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ij} = \sum_{j=1}^{n}\sum_{i=1}^{N} u_{ij} + \sum_{j=n+1}^{m}\sum_{i=1}^{N} u_{ij}. \tag{6.4.6}$$

For visualizing purpose, we define U to be a matrix where the elements are u_{ij}.

$$U = \left[\begin{array}{ccc|ccc} u_{11}=1 & \dots & u_{n1} & u_{1(n+1)} & \dots & u_{1m} \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ u_{n1} & \dots & u_{nn}=1 & & & \\ \hline u_{(n+1)1} & \dots & u_{(n+1)n} & & & \\ \vdots & & \vdots & \vdots & & \vdots \\ u_{N1} & \dots & u_{Nn} & u_{N(n+1)} & \dots & u_{NM} \end{array}\right].$$

The matrix U will be used to break down Eqn (6.4.6) to ease the proof strategy. Based on matrix U, we can further split Eqn 6.4.6 into

$$\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ij} = \sum_{\substack{i,j=1\\ i=j}}^{n} u_{ij} + \sum_{\substack{i,j=1\\ i\neq j}}^{n} u_{ij} + \sum_{j=1}^{n}\sum_{i=n+1}^{N} u_{ij} + \sum_{j=n+1}^{m}\sum_{i=1}^{N} u_{ij}. \tag{6.4.7}$$

The first term and the second term in Eqn 6.4.7 refers to the sum of diagonal elements and the sum of off-diagonal elements of of upper left block of matrix U. The third and fourth term of Eqn 6.4.7 refer to sum of elements of the lower left and the sum of elements of the right (lower and upper) block of matrix U, respectively. We will compute the limit of each term of Eqn 6.4.7 as $h \to 0$ individually.

Firstly, for the first term of Eqn 6.4.7. The diagonal entry of each term is 1, this leads to

$$\lim_{h\to 0}\sum_{\substack{i,j=1\\ i=j}}^{n} u_{ij} = n \tag{6.4.8}$$

because there are n diagonal entries. This follows from the elementary property of exponential function which is $\mathrm{e}^0 = 1$. For the second term of Eqn 6.4.7, the limit as $h \to 0$ is

$$\lim_{h\to 0}\sum_{\substack{i,j=1\\ i\neq j}}^{n} u_{ij} = 0. \tag{6.4.9}$$

This follows from Lemma C.2.3.

For the third and fourth terms of Eqn 6.4.7, the limits as $h \to 0$ are

$$\lim_{h\to 0}\sum_{j=1}^{n}\sum_{i=n+1}^{N} u_{ij} =0 \tag{6.4.10}$$

$$\lim_{h\to 0}\sum_{j=1}^{m}\sum_{i=1}^{N} u_{ij} =0 \tag{6.4.11}$$

respectively, which also follow from Lemma C.2.3. Then, the limit as $h \to 0$ for Eqn (6.4.7) is

$$\lim_{h\to 0}\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ij} = \lim_{h\to 0}\sum_{\substack{i,j=1\\ i=j}}^{n} u_{ij} + \lim_{h\to 0}\sum_{\substack{i,j=1\\ i\neq j}}^{n} u_{ij} + \lim_{h\to 0}\sum_{j=1}^{n}\sum_{i=n+1}^{N} u_{ij} + \lim_{h\to 0}\sum_{j=n+1}^{m}\sum_{i=1}^{N} u_{ij} \tag{6.4.12}$$

$$=n. \tag{6.4.13}$$

Multiply by $-4N$,

$$-4N\lim_{h\to 0}\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ij} = -4Nn. \tag{6.4.14}$$

Therefore, as $h \to 0$, the limit of the first term of the numerator in Eqn (6.4.5) tends to $-4Nn$.

Now, we focus on the second term of the numerator of Eqn 6.4.5. For visualization purposes, we define W to be a matrix where the elements are w_{ij} for $i = 1, \ldots, N$ and $j = 1, \ldots, M$.

$$W = \begin{bmatrix} w_{11} = 1 & \ldots & \ldots & w_{1N} \\ \vdots & & & \vdots \\ w_{n1} & \ddots & & w_{nN} \\ \vdots & & & \vdots \\ w_{N1} & \ldots & \ldots & w_{NN} = 1 \end{bmatrix}.$$

We express the summation of the second term of the numerator of Eqn 6.4.5 as

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} = \sum_{\substack{i,j=1 \\ i=j}}^{N} w_{ij} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N} w_{ij}. \tag{6.4.15}$$

The first term of Eqn 6.4.15 refers to the summation of the diagonal entry of W while the second term refers to the summation of the off-diagonal entry of matrix W.

Consider the first term of Eqn (6.4.15). As $h \to 0$, the limit of $\sum_{\substack{i,j=1 \\ i=j}}^{N} w_{ij}$ is

$$\lim_{h \to 0} \sum_{\substack{i,j=1 \\ i=j}}^{N} w_{ij} = N. \tag{6.4.16}$$

The diagonal entries consist of the terms $Y_i = Y_j$ for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$. This results to $Y_i - Y_j = 0$. Therefore, by the elementary property of exponential function, $\mathrm{e}^0 = 1$. Hence, each diagonal entry of matrix W is 1 and the sum of the diagonal entry in N.

For the second term Eqn 6.4.15, the limit of $\sum_{\substack{i,j=1 \\ i \neq j}}^{N} w_{ij}$ as $h \to 0$, is

$$\lim_{h \to 0} \sum_{\substack{i,j=1 \\ i=j}}^{N} w_{ij} = 0. \tag{6.4.17}$$

This follows from Lemma C.2.3.

Therefore, the limit of the summation of the second term of the numerator of Eqn

6.4.5 as $h \to 0$ is

$$\lim_{h\to 0} \sum_{j=1}^{N} \sum_{i=1}^{N} w_{ij} = N. \tag{6.4.18}$$

Multiply by $M\sqrt{2}$, we obtain

$$M\sqrt{2} \lim_{h\to 0} \sum_{j=1}^{N} \sum_{i=1}^{N} w_{ij} = M\sqrt{2}N. \tag{6.4.19}$$

Therefore, the limit the second term of the numerator of Eqn (6.4.5) is $M\sqrt{2}N$ as $h \to 0$.

Then, adding Eqn (6.4.14) and Eqn (6.4.19) we obtain

$$\lim_{h\to 0} g(h) = \lim_{h\to 0} \frac{1}{h} . \frac{-4Nn + NM\sqrt{2}}{2MN^2\sqrt{2\pi}} \tag{6.4.20}$$

$$= \lim_{h\to 0} \frac{1}{h} . \frac{-4n + M\sqrt{2}}{2MN\sqrt{2\pi}} . \tag{6.4.21}$$

Now, we can show the limit of Eqn (6.4.5) for case (1) $M > 2\sqrt{2}n$ and (2) $M < 2\sqrt{2}n$.

1. $\boldsymbol{M > 2\sqrt{2}n}$: Using the assumption $M > 2\sqrt{2}n$, we show that

$$\lim_{h\to 0} g(h) > \lim_{h\to 0} \frac{1}{h} . \frac{-4n + M\sqrt{2}}{2(2\sqrt{2})N\sqrt{2\pi}} \tag{6.4.22}$$

$$= \lim_{h\to 0} \frac{1}{h} . \frac{-4n + M\sqrt{2}}{8Nn\sqrt{\pi}} \tag{6.4.23}$$

$$= \lim_{h\to 0} \frac{1}{h} . \frac{-4 + \frac{M}{n}\sqrt{2}}{8N\sqrt{\pi}} \tag{6.4.24}$$

$$= \lim_{h\to 0} \frac{1}{h} . \frac{-2\sqrt{2} + \frac{M}{n}}{4\sqrt{2}N\sqrt{\pi}} \tag{6.4.25}$$

$$= \lim_{h\to 0} \frac{1}{h} . \frac{-2\sqrt{2} + \frac{M}{n}}{4N\sqrt{2\pi}} \tag{6.4.26}$$

Since $M > 2\sqrt{2}n \implies \frac{M}{n} > 2\sqrt{2}$, the numerator of Eqn (6.4.26) will always be positive and never equal to 0. Since $\lim\limits_{h\to 0} \frac{1}{h} = \infty$, by the property of limit,

$\lim\limits_{h\to 0} C.\frac{1}{h} = \infty$. Taking $C = \frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}}$,

$$\lim_{h\to 0} \frac{1}{h}.\frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}} = \infty. \tag{6.4.27}$$

Since $g(h) > \frac{1}{h}.\frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}}$, then $\lim\limits_{h\to 0} g(h) = \infty$. Therefore, we prove that $\lim\limits_{h\to 0} g(h) = \infty$ when $M > 1\sqrt{2}n$.

2. $\boldsymbol{M < 2\sqrt{2}n}$: Using the assumption $M < 2\sqrt{2}n$, we show that

$$\lim_{h\to 0} g(h) < \lim_{h\to 0} \frac{1}{h}.\frac{-4n+M\sqrt{2}}{2(2\sqrt{2})N\sqrt{2\pi}} \tag{6.4.28}$$

$$= \lim_{h\to 0} \frac{1}{h}.\frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}} \tag{6.4.29}$$

When $M < 2\sqrt{2}n \implies \frac{M}{n} \leq 2\sqrt{2}$, the numerator of Eqn (6.4.29 will always be negative and never 0. Since $\lim\limits_{h\to 0} \frac{1}{h} = \infty$, by the property of limit, $\lim\limits_{h\to 0} C.\frac{1}{h} = \frac{1}{h}.C$. Taking $C = \frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}}$, we can show that the limit of the RHS of Eqn (6.4.29) is negative because is C is always negative. Since $g(h) < \frac{1}{h}.\frac{-2\sqrt{2}+\frac{M}{n}}{4N\sqrt{2\pi}}$, $\lim\limits_{h\to 0} = -\infty$ as $h \to 0$ under $M < 2\sqrt{2}n$.

□

Proposition 6.4.1. *Define the function $g : \mathbb{R}^+ \to \mathbb{R}$ as in Eqn (6.4.1). Then for M and n from Section 6.4.1.1,*

1. *when $M > 2\sqrt{2}n$, g tends to 0 as $h \to \infty$, i.e.*

$$\lim_{h\to\infty} g(h) = 0. \tag{6.4.30}$$

2. *when $M < 2\sqrt{2}n$, g tends to 0 as $h \to \infty$, i.e.*

$$\lim_{h\to\infty} g(h) = 0. \tag{6.4.31}$$

Proof. Recall Eqn (6.4.1). For convenience, we re-write the equation again,

$$g(h) = -\frac{2}{NM\sqrt{2\pi}} \sum_{j=1}^{M}\sum_{i=1}^{N} \frac{1}{h} \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)} + \frac{1}{2N^2\sqrt{\pi}} \sum_{j=1}^{N}\sum_{i=1}^{N} \frac{1}{h} \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)^2}. \tag{6.4.32}$$

We proof this by using the limit of exponential. Using Lemma (C.2.2) ,

$$\lim_{h\to\infty} \frac{1}{h} \mathrm{e}^{-\frac{1}{2}\left(\frac{t_{ij}}{h}\right)} = 0$$

and

$$\lim_{h\to\infty} \frac{1}{h} \mathrm{e}^{-\frac{1}{4}\left(\frac{r_{ij}}{h}\right)} = 0.$$

Therefore,

$$\lim_{h\to\infty} g(h) = -\frac{2}{NM\sqrt{2\pi}} \sum_{j=1}^{M}\sum_{i=1}^{N} 0 + \frac{1}{2N^2\sqrt{\pi}} \sum_{j=1}^{N}\sum_{i=1}^{N} 0. \tag{6.4.33}$$

Hence, $\lim\limits_{h\to\infty} g(h) = 0$. Therefore, we show that the limit of Eqn (6.4.1) for

1. $M > 2\sqrt{2}$

$$\lim_{h\to\infty} g(h) = 0 \tag{6.4.34}$$

2. $M < 2\sqrt{2}$

$$\lim_{h\to\infty} g(h) = 0. \tag{6.4.35}$$

□

6.4.2.1 Proof for a Minima for PSL

In this section, we provide proof to show that when $M > 2\sqrt{2}n$ for M and n from Section 6.4.1.1, there exist a minimum point for out-of-sample empirical PSL.

Lemma 6.4.3. *Let $g : \mathbb{R}^+ \to \mathbb{R}$ be the function defined in Def 6.4.1. Let M and n from Section 6.4.1.1. When*

1. *$M > 2\sqrt{2}n$, $\lim\limits_{h\to 0} g(h) = \infty$*
2. *$M < 2\sqrt{2}n$, $\lim\limits_{h\to\infty} g(h) = 0$*
3. *There exist $L \in \mathbb{R}$ such that $\forall h > 0 : g(h) \geq L$*

Proof. 1. See 1 of Lemma 6.4.2

2. See 1 of Proposition 6.4.1

3. Let $c_1 \in \mathbb{R}^+$. Suppose there is $h_1 > 0$, then $g(h_1) > 0$. From 1 of Lemma 6.4.2, there is $h_2 > 0$ for all $h < h_2$ such that $g(h) > c_1$. Following 1 of Proposition

6.4.1, there is $h_3 > 0$ for all $h > h_3$ such that $g(h) > c_1$. Let $\tilde{h}_2 = \min(h_2, h_3)$ and $\tilde{h}_3 = \max(h_2, h_3)$. Also, we let $g(h_1) = c_1$. We know that $g(h) > c_1$ if $h > \tilde{h}_3$ or $h < \tilde{h}_2$. Then we have $g(h) \leq c_1$ when $h \in [\tilde{h}_2, \tilde{h}_3]$ where the compact interval $I = [\tilde{h}_2, \tilde{h}_3]$ will always be none empty because h_1 will always be in I. Then, we can show that $\forall h \notin I : g(h) > c_1$ and $\exists h \in I : g(h) \leq c_1$. By Lemma 6.3.5, for any interval $I = [\tilde{h}_2, \tilde{h}_3]$, $g(I) = \{g(h) : h \in I\}$ is a compact set and we know that there exist a minimum in the compact set $f(I)$. We also know that $g(h) > \min g(I) = L$ if $h \in I$. Hence, we can conclude that $g(h) \geq \min\left(0, \min g(I)\right)$ for any $h > 0$.

□

6.4.2.2 Theorem on Global Minimum of Out-of-sample Empirical PSL of Gaussian Kernel pdf

In this final theorem below, we provide a proof in the case where $M > 2\sqrt{2}n$ for M and n from Section 6.4.1.1, there exist a minimum and that minimum point is the global minimum for out-of-sample empirical PSL of Gaussian kernel.

Theorem 6.4.1. *Let g be defined in Def 6.4.1. Let M and n from Section 6.4.1.1. Then, the followings are equivalent.*

1. *$M > 2\sqrt{2}n$*
2. *g has a global minimum .*

Proof. Here, we present the proof in two directions:

1. (1) → (2).
   If $M > 2\sqrt{2}n$, then g has a global minimum
2. (2) → (1) by using contraposition, $\neg(1) \rightarrow \neg(2)$
   If $M < 2\sqrt{2}n$ then g has no global minimum.

We construct the first proof. By Lemma 6.4.1, g is a continuous function. Suppose there is $h_1 > 0$, then $g(h_1) > 0$. From 1 of Lemma 6.4.2, there is $h_2 > 0$ for all $h < h_2$ such that $g(h) > c_1$. Following 1 of Proposition 6.4.1, there is $h_3 > 0$ for all $h > h_3$ such that $g(h) > c_1$. Let $\tilde{h}_2 = \min(h_2, h_3)$ and $\tilde{h}_3 = \max(h_2, h_3)$. Also, we let $g(h_1) = c_1$. We know that $g(h) > c_1$ if $h > \tilde{h}_3$ or $h < \tilde{h}_2$. Then we have $g(h) \leq c_1$ when $h \in [\tilde{h}_2, \tilde{h}_3]$ where the compact interval $I = [\tilde{h}_2, \tilde{h}_3]$ will always be none empty because h_1 will always be in I. Then, we can show that $\forall h \notin I : g(h) > c_1$ and $\exists h \in I : g(h) \leq c_1$. By Lemma 6.3.5, for any interval

$I = [\tilde{h}_2, \tilde{h}_3]$, $g(I) = \{f(h) : h \in I\}$ is a compact set. Hence, by extreme value theorem, there exist a minimum when f is continuous on the compact set I. Since $\forall h \notin I : g(h) > c_1$ we know that there is no h outside of I will give $g(h) < c_1$, hence the minimum in the compact set I is also the global minimum.

For the second proof, we prove this by contraposition, such that if $M < 2\sqrt{2}n$, then g has no global minimum. This is easily seen by 2 of Lemma 6.4.2 and 2 of Proposition 6.4.1, where when $M < 2\sqrt{2}n$, $\lim_{h \to 0} g(h) = -\infty$ and $\lim_{h \to \infty} g(h) = 0$, respectively. Hence, there is no global minimum when $M < 2\sqrt{2}n$ □

6.5 Experiment

In this section, we run a simulation experiment to compare the difference in the behaviour of in-sampling tuning and out-of-sample tuning using grid search for univariate Gaussian kernel PDF as the estimator via log-loss and PSL. This experiment will be conducted on 2 simulated datasets, the Old Faithful Geyser and 3 datasets from UCI [111].

From this simulation experiment, the results using out-of-sample empirical log-loss will achieve a minimum point resulting to an optimum bandwidth. However, when using the out-of-sample empirical PSL for tuning, some the datasets will achieve a minimum point while some datasets shows an increasing pattern. This is due to the number of test data points that exist in both training fold and test folds to be less than the proposed ratio for Gaussian kernel. Furthermore, the bandwidth selected via in-sample empirical log-loss and out-of-sample empirical PSL with many repeated data points results to a loss when evaluated.

6.5.1 Objective of Simulation Experiment

The objective of this experiment is compare the behaviour of in-sample and out-of-sample empirical loss for tuning. The experiments uses univariate Gaussian kernel PDF as the estimators. The loss functions used are log-loss and PSL. The tuning algorithms is done via grid search. We will then evaluate the tuned model using out-of-sample empirical log-loss to compare the results between in-sample tuned method and out-of-sample tuned method.

6.5.2 Design of Experiment

We describe the design of the experiment including datasets used, the estimator, resampling method for tuning and evaluation.

Datasets: The datasets that we consider includes both simulated datasets and datasets from uci ([111]).

1. **Dataset 1:** A simulated dataset of 200 data points from a Normal distribution with mean 2 and standard deviation 1.
2. **Dataset 2:** A simulated dataset of 200 data points from a mixture of Normal distribution with mean 1 and standard deviation 1 and Normal distribution with mean 10 and standard deviation 1.
3. **Boston Housing:** The dataset concerns with the value of the houses in Boston. It has 13 features and a label variables with 506 data points for each variable. The variable that we will use for this tuning experiment is the "Median value of owner-occupied homesin $1000".
4. **Old faithful Geyser:** This is the most common used dataset for investigating distribution estimation. The total number of data points is 272 and we use the variable "Duration of eruptions" in this experiment.
5. **Energy efficiency:** The dataset is used to predict the heating and cooling loads requirement of a building. It has 768 data points and the variable used for this experiment is "cooling load".
6. **Auto mpg:** This data concerns with the city-cycle fuel consumption that is measured in miles per gallon. The dataset consists of 398 data points. We focus the variable "mpg" for this experiment.

Resampling and Evaluating: To avoid bias when evaluating the goodness of the selected model, the training set should be different from the ones used for evaluation. Therefore, each dataset is split using 3-fold cross-validation. Each of the tuned model will be fitted on its training set and evaluated on the test set. In this experiment, we use the log-loss to evaluate and the computation of the empirical loss will follow Algorithm 17. Since we are using 3-fold cross-validation, we expect to obtain 3 empirical log-loss. The average over the 3-fold is computed and we call this as the 'mean empirical loss'. This is the value that will be reported.

Estimator: For this experiment, we focus only on the univariate Gaussian kernel PDF, f as in Eqn (2.3.4) where the Gaussian kernel is

$$K(u) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{u^2}{2}}. \tag{6.5.1}$$

Then, the out-of-sample estimator at each point Y^* is

$$f(Y^*) = \frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} \mathrm{e}^{\frac{-(Y^*-Y_i)^2}{2h^2}} \tag{6.5.2}$$

Parameter for Tuning methods: The parameter that will be tuned is the bandwidth. In this tuning method, we will use grid search. A vector of values for bandwidth is pre-specified:

$$\begin{aligned} h = &(1.5863158, 6.0000000, 0.6405263, 3.7931579, 1.9015789, 4.7389474, 4.1084211, \\ &5.6847368, 5.3694737, 2.2168421, 0.0100000, 3.1626316, 2.8473684, 2.5321053, \\ &4.4236842, 0.9557895, 3.4778947, 5.0542105, 0.3252632, 1.2710526) \end{aligned}$$

The estimator in Eqn (6.5.2) is learned on this values. The bandwidth with the smallest empirical loss will be selected as the tuned model. The range of bandwidth for this method is

Resampling for Tuning: There are two two re-sampling methods for tuning we will use for this experiment. Since tuning occurs in the fitting, we will mainly discussed the splitting of the training dataset here.

1. **In-sampling:** For this method, the same training dataset is used for *inner* training and predicting. No further resampling is done. This will follow method explained in Figure 6.4.
2. **Out-of-sample:** For this method, the training dataset is further split into *inner training dataset* and *inner test dataset* using hold out method. The distribution (PDF) is estimated on the *inner test dataset* using the *inner training set* as s observations. The inner prediction is evaluated and the empirical log-loss is computed. For grid search method, this whole process is repeated for each bandwidth h_b for $b = 1, \ldots, B$. The bandwidth with the minimum empirical loss is selected as the tuned bandwidth which reflects as the tuned Gaussian kernel PDF model. The algorithm will follow from Figure 6.3 and Algorithm 18.

6.5.3 Results

In this section, we discuss the result of the experiment to compare the behaviour of the empirical loss for tuning the bandwidth. We use two methods to tune the bandwidth, in-sample and out-of-sample tuning methods. The experiment is done on 6 datasets via grid-search using Gaussian kernel PDF. We first explain the result of tuning via log-loss and later via PSL.

Results Using Log-loss: The plots of empirical log-loss against log bandwidth for out-of-sample and in-sample are shown in Figure 6.5 and Figure 6.6, respectively. These results are obtained during the tuning stage. In each graph, we plot the empirical log-loss against log of bandwidth for each fold. The results are expected and agreed with the proof in Section 6.3.2. The out-of-sample log-loss for all datasets show a minimum point for each fold. The in-sample empirical log-loss for all dataset show increasing pattern as the log bandwidth increase (decreasing flexibility).

Table 6.1 shows the optimal (tuned) bandwidth for each dataset and each fold. Table 6.2 shows the minimum out-of-sample empirical log-loss reflecting to the optimal bandwidth. It can be seen that the bandwidth obtained by out-of-sample tuning method varies for each fold. However, this not the case for in-sample tuning method. Using the in-sample tuning method, each fold output the same tuned bandwidth for each dataset. This shows that in-sample tuning will select the bandwidth that increases the flexibility, that is the smallest bandwidth. This shows that overfitting occur during the in-sample tuning method.

Once the tuned method is selected, they are evaluated on the *outer* folds. The results of the evaluation using out-of-sample empirical generalization log-loss are shown in Table 6.3 where the mean empirical losses are reported. The mean empirical loss for out-of-sample tuning methods are smaller than then in-sample tuned methods.

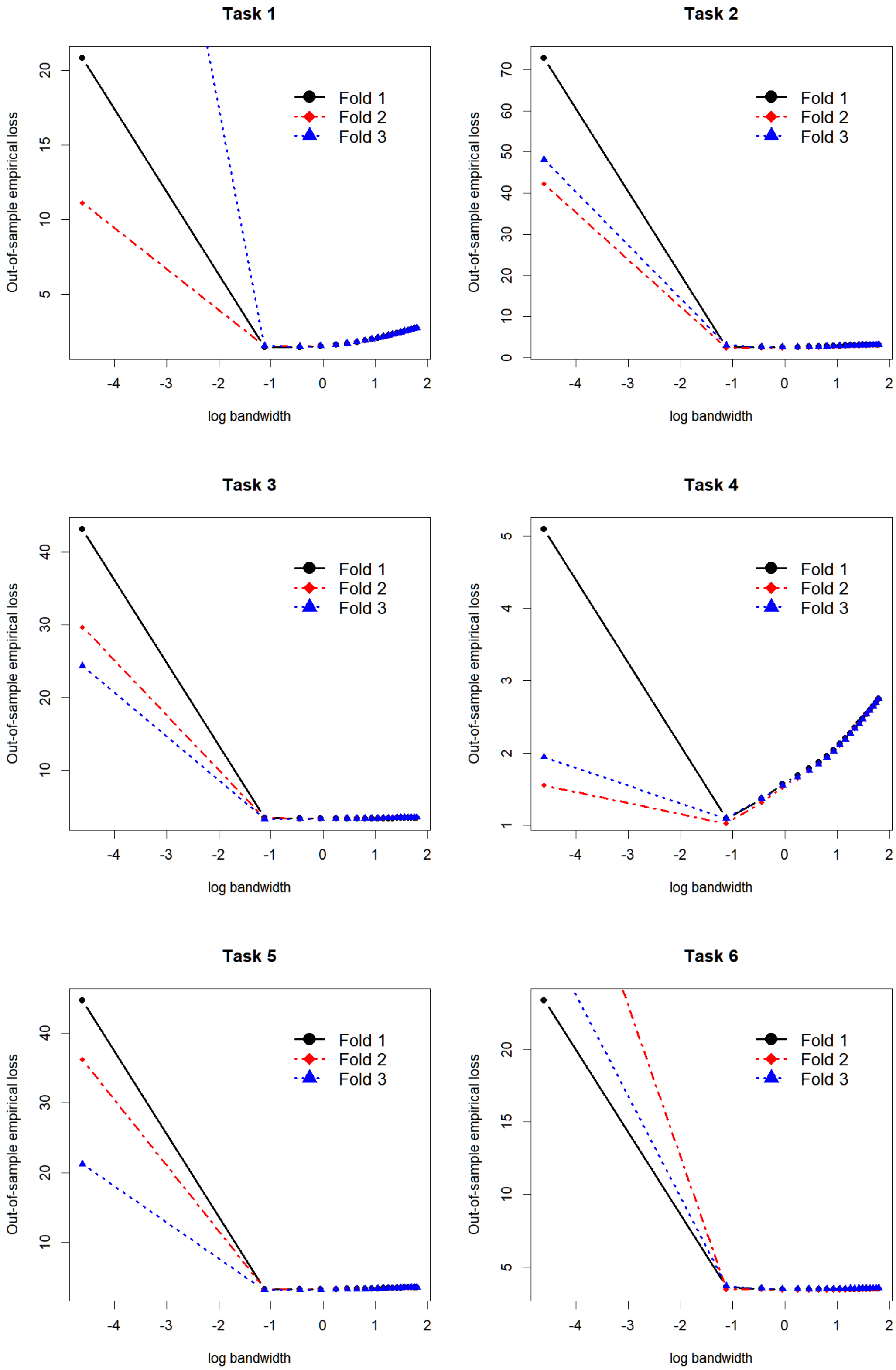


Figure 6.5: Plots for out-of-sample tuning using log-loss against log bandwidth for each dataset. The y-axis is the out-of-sample empirical log-loss and the x-axis is the log bandwidth. The black line refers to the first fold, red line refers to the second fold and the blue line refers to the third fold. (Task 1: Dataset 1 , Task 2: Dataset 2; Task 3: Boston, Task 4: Old Faithful; Task 5: Energy; Task 6: Auto)

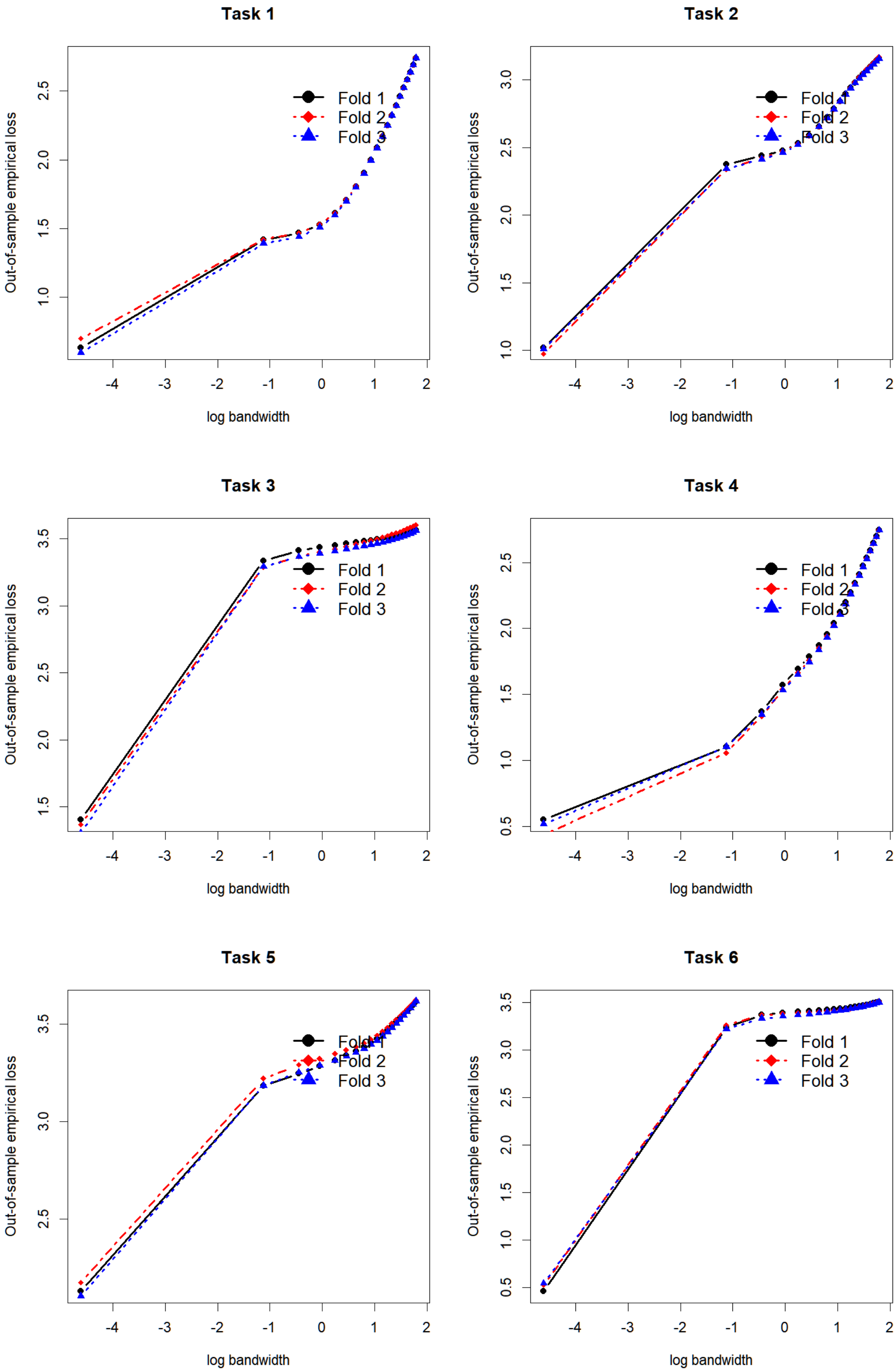


Figure 6.6: Plots for in-sample tuning using log-loss against log bandwidth for each dataset. The y-axis is the out-of-sample empirical log-loss and the x-axis is the log bandwidth. The black line refers to the first fold, red line refers to the second fold and the blue line refers to the third fold. (Task 1: Dataset 1, Task 2: Dataset 2; Task 3: Boston, Task 4: Old Faithful; Task 5: Energy; Task 6: Auto)

Dataset	Bandwidth					
	Out-of-sample			In-sample		
	Fold 1	Fold 2	Fold 3	Fold 1	Fold	Fold 3
Data 1	0.3253	0.3253	0.6405	0.1000	0.1000	0.1000
Data 2	0.6405	0.3253	0.9558	0.1000	0.1000	0.1000
Boston	1.2711	0.6405	0.3253	0.1000	0.1000	0.1000
Old faithful	0.3253	0.3253	0.3253	0.1000	0.1000	0.1000
Energy	0.6405	0.3253	0.3253	0.1000	0.1000	0.1000
Auto	1.9016	1.5863	1.5863	0.1000	0.1000	0.1000

Table 6.1: Table of tuned bandwidth via log-loss obtained from training in each fold. For out-of-sample, the result is from using *outer* training set whereas form in-sample, the result is from using the same data for fitting and predicting.

Dataset	Out-of-sample Empirical Log-loss					
	Out-of-sample			In-sample		
	Fold 1	Fold 2	Fold 3	Fold 1	Fold	Fold 3
Data 1	1.4515	1.5403	1.4982	0.6342	0.7013	0.5999
Data 2	2.5558	2.4148	2.5677	1.0195	0.9761	1.0103
Boston	3.4113	3.4326	3.4326	1.4030	1.3665	1.3056
Old faithful	1.0910	1.0217	1.0928	0.5491	0.4359	0.5178
Energy	3.3102	3.3433	3.1798	2.1289	2.1743	2.1042
Auto	3.4593	3.4251	3.4755	0.4608	0.5237	0.5431

Table 6.2: Table of empirical log-loss for in-sample and out-of-sample methods during tuning. The out-of-sample shows the result of the empirical log-loss obtained by using different folds for fitting and predicting. The in-sample result shows the empirical log-loss obtained by using the same fold for fitting and predicting.

Dataset	Mean Empirical Log-loss	
	Out-of-sample	In-sample
Data 1	1.446286	7.015544
Data 1	2.475117	22.66495
Boston	3.493477	31.97605
Old faithful	1.113645	3.320471
Energy	3.300547	18.15904
Auto	3.424956	27.87395

Table 6.3: Table of mean empirical log-loss obtained by evaluating each tuned model via log-loss.

Results Using PSL: The plots of empirical PSL against log bandwidth for out-of-sample and in-sample are shown in Figure 6.7 and Figure 6.8, respectively. These results are obtained during the tuning stage. In each graph, we plot the empirical PSL against log of bandwidth for each fold. The out-of-sample empirical PSL for all datasets except Boston and Auto show existence of a minimum point for each fold. The in-sample empirical PSL for all dataset show increasing pattern as the log bandwidth increase (decreasing flexibility).

For PSL, the results from tuning via in-sample and out-of-sample is shown in Table 6.4 and Table 6.5. Table 6.4 shows the bandwidth obtained in each fold when tuning via in-sample and out-of-sample. For in-sample tuning, the result is consistent to the result when using in-sample tuning via log-loss. For datasets Boston and Auto both, the bandwidth selected from using out-of-sample tuning and in-sample tuning are the same for all folds, which is 0.100.

Once the tuned bandwidth is selected, they are evaluated using the *outer* folds via log-loss. The results of the evaluation using out-of-sample empirical log-loss are shown in Table 6.6. For datasets Data 1, Data 2, Old faithful and Energy, the mean empirical log-loss for in-sampled tuned method are larger than the out-of-sample tuned methods. However, this is not the case for Boston and Auto. The mean empirical log-loss for in-sampled tuned method and out-of-sample are the same. This is because there are multiple repeated observations in both datasets. There is a tendency that the *outer* does not exceed the test to training data ratio.

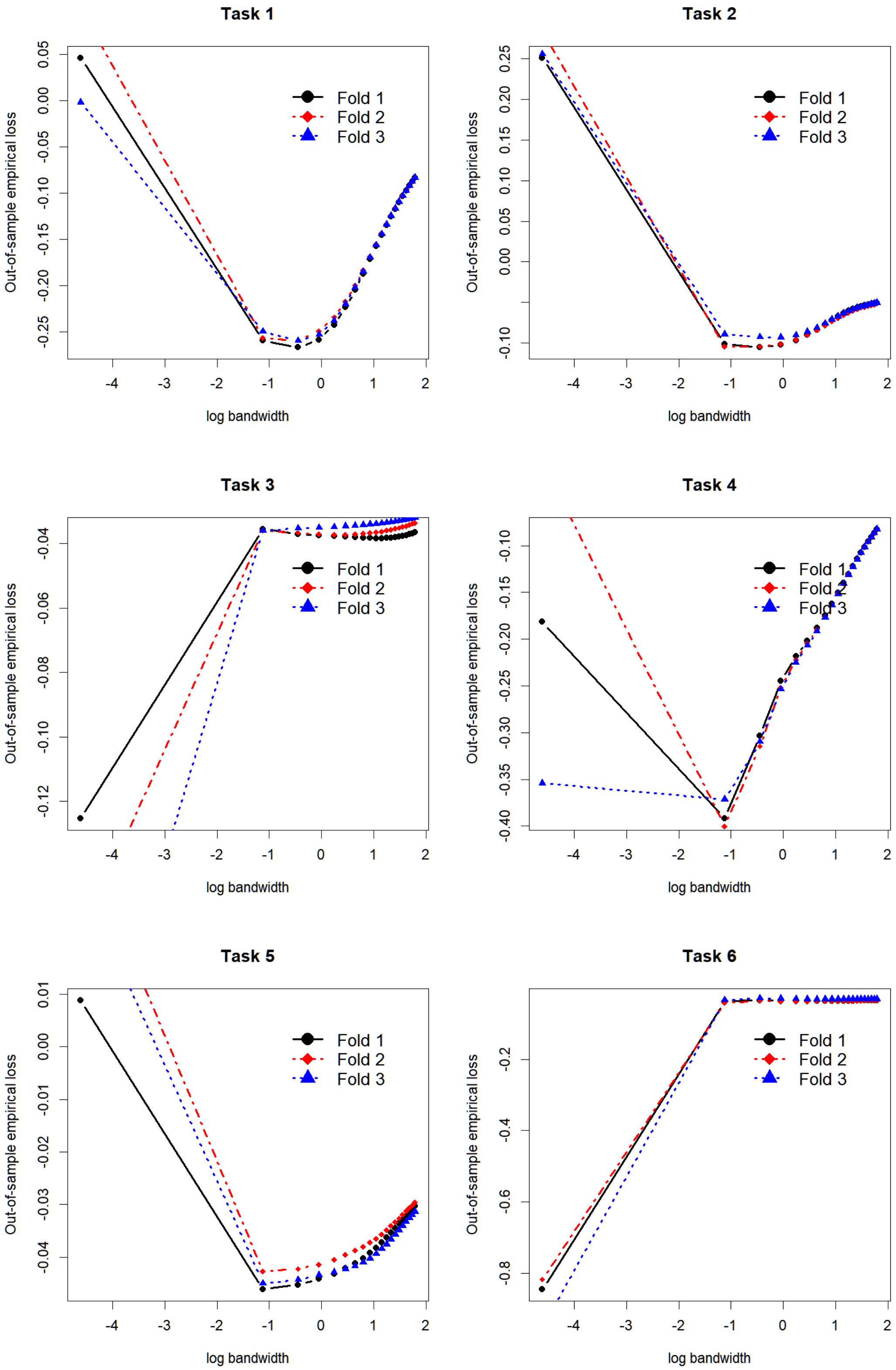


Figure 6.7: Plots for out-of-sample tuning using PSL against log bandwidth for each dataset. The y-axis is the out-of-sample empirical PSL and the x-axis is the log bandwidth. The black line refers to the first fold, red line refers to the second fold and the blue line refers to the third fold. (Task 1: Dataset , Task 2: Dataset 2; Task 3: Boston, Task 4: Old Faithful; Task 5: Energy; Task 6: Auto)

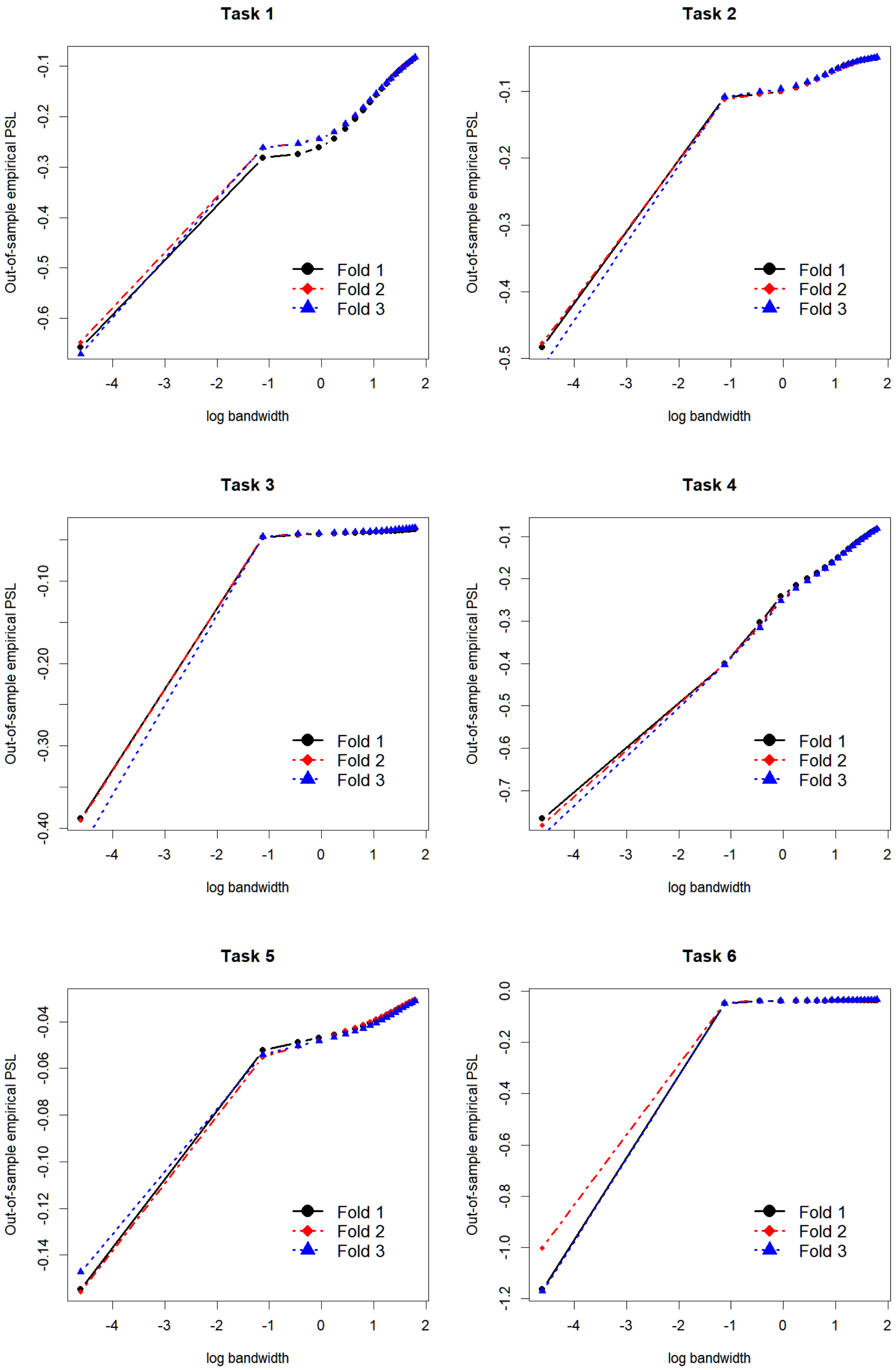


Figure 6.8: Plots for in-sample tuning using PSL against log bandwidth for each dataset. The y-axis is the out-of-sample empirical PSL and the x-axis is the log bandwidth. The black line refers to the first fold, red line refers to the second fold and the blue line refers to the third fold. (Task 1: Dataset , Task 2: Dataset 2; Task 3: Boston, Task 4: Old Faithful; Task 5: Energy; Task 6: Auto)

Dataset	Bandwidth					
	Out-of-sample			In-sample		
	Fold 1	Fold 2	Fold 3	Fold 1	Fold	Fold 3
Data 1	0.6405	0.6405	0.6405	0.1000	0.1000	0.1000
Data 2	0.6405	0.6405	0.9558	0.1000	0.1000	0.1000
Boston	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
Old faithful	0.3253	0.3253	0.3253	0.1000	0.1000	0.1000
Energy	0.3253	0.3253	0.3253	0.1000	0.1000	0.1000
Auto	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000

Table 6.4: Table of results of tuned bandwidth obtained using PSL. The out-of-sample result is from training in each fold using *outer* training set. The in-sample result is from using the same fold for training and predicting.

Dataset	Out-of-sample Empirical PSL					
	Out-of-sample			In-sample		
	Fold 1	Fold 2	Fold 3	Fold 1	Fold	Fold 3
Data 1	-0.2665	-0.2587	-0.2594	-0.6576	-0.6475	-0.6713
Data 2	-0.1050	-0.1043	-0.0928	-0.4833	-0.4772	-0.5129
Boston	-0.1253	-0.1619	-0.2248	-0.3883	-0.3901	-0.4253
Old faithful	-0.3919	-0.4004	-0.3713	-0.7661	-0.7820	-0.8065
Energy	-0.0461	-0.0428	-0.0450	-0.1547	-0.1556	-0.1472
Auto	-0.8446	-0.8166	-0.9500	-1.1634	-1.0011	-1.1700

Table 6.5: Table empirical PSL for in-sample and out-of-sample tuning methods. The out-of-sample column shows the result of the empirical PSL obtained using different folds for fitting and predicting. The in-sample result shows the empirical PSL obtained by using the same fold for fitting and predicting.

Dataset	Mean Empirical Log-loss	
	Out-of-sample	In-sample
Data 1	1.5051	12.0565
Data 1	2.5350	26.1597
Boston	26.1597	37.6821
Old faithful	1.1061	3.5684
Energy	3.2751	15.892
Auto	29.3960	29.3960

Table 6.6: The mean empirical log-loss obtained by evaluating each tuned model via PSL.

6.6 Discussion & Conclusion

In this chapter, we investigate the behaviour between the in-sample and out-of-sample tuning for univariate distribution estimation using the Gaussian kernel via log-loss and PSL. For log-loss, the proof and simulation result show that in-sample tuning results to an unbounded function in which as the bandwidth goes to 0 and ∞ the in-sample empirical loss tends to $-\infty$ and 0, respectively. However, we discovered that for the out-of-sample empirical log-loss to achieve a minimum point, only one new unobserved point in the test set is needed. In this case, as the bandwidth goes to 0 and ∞ the out-of-sample empirical log-loss goes to ∞. This is supported by [58] that states if the point at which PDF is estimated is equal to the sample data, it will reach infinity (for likelihood cross-validation).

When tuning with PSL, [5] stated that discretization of data (even if dataset is from a continuous random variable, in real life we can only limit the decimal points) will cause out-of-sample empirical PSL to go from ∞ to 0 as the bandwidth tends to 0 and ∞, respectively. This however can be overcomed when the test data and training data reach a threshold. We proved that for out-of-sample empirical PSL to be bounded when using Gaussian kernel PDF, the ratio of size of the test set to the number of repeated data points in both test set and training set should be greater than $2\sqrt{2} : 1$. Our prove is not only to support statement by [5] but also clearly differentiate the use of training set and test set for out-of-sample tuning. Under this condition, a global minimum is achieved.

Chapter 7

A Unified Interface for Distribution Estimation in R

7.1 Introduction

This chapter focusses on the investigation and implementation of a unified machine learning (ML) interface for distribution estimation. There are three key elements that we discuss in this chapter. Firstly, we review the existing machine learning (ML) toolbox and where distribution estimation sits in the toolbox. Secondly, we explain the integration of distribution estimation into an existing toolbox in **R** ([112]) which is **mlr3proba** [113]. Lastly, we explain the relationship between distribution packages, **distr6** [114] and **mlr3proba**. The objective is to implement a unified ML interface for distribution estimation.

With the development of computer software, the computation of mathematics and statistical learning are made easier. Developers implement mathematics and statistics methods in the software which can later be used by other users and for different purposes. Some software that have already implemented a unified machine learning interface (training, predicting and evaluating) - **scikit-learn** [115] in Python [116], **weka** [117] in Java [118] and **mlr3** [119], **mlr** [120], **tinymodel** [121] and **caret** [122] in **R**. **R** which was developed for statistical learning contains many packages with functions to support different statistical methods including distribution estimation. However, there are no specific packages or tools that support a unified machine learning interface (training, predicting and evaluating) for distribution estimation in **R**.

The purpose for a unified interface for distribution estimation is to provide users an easy and straight-forward implementation of distribution estimation using a machine learning framework without the hassle to compute from scratch. With this platform, users are able to train/fit distribution learner, predict the distribution, (PDF and (or) CDF) and also evaluate the distribution model using different loss functions.

In this chapter, we discuss the unified interface for distribution estimation. This interface uses the existing **mlr3** ecosystem, specifically **mlr3proba** and **mlr3extralearners** and also **distr6**. **mlr3** provides interface for regression and classification whereas **mlr3proba** currently supports one of probabilistic machine learning which is survival analysis. Since distribution estimation is a probabilistic learning, we incorporate interface for distribution estimation within **mlr3proba**.

The outline this chapter is as follows. Section 7.2 explores the existing machine learning (ML) toolbox. Section 7.3 explains the motivation of the unified ML interface for distribution and a review on distribution related packages in **R**. Section 7.4 is a more detail review on **distr6** and we explain our contribution to the package. Then, in Section 7.5 will focus on the design of the unified ML interface for distribution estimation in which we explain the connection between **mlr3**, **mlr3proba**, **mlr3extralearners** and **distr6**.

Terminology

Before we proceed, we first need to explain the terms used throughout this chapter for clarity.

i. **Mathematical or statistical software:** An application that is used to solved problems relating to mathematics and statistics. Problems may include solving numerical computation, modelling, producing graphs and etc.
ii. **Software package or toolbox:** A collection of functions (may also include datasets) that is useful to target a specific problem. For example, the package **Matrix** in **R** contains various functions that are specific to solve problems regarding matrices.
iii. **Functional programming**: Solving problems in by defining functions. Data is not stored as an object.
iv. **Object oriented (OO) programming:** A programming method that focusses on *object* and *class*. **R** has multiple OO programming languages (e.g. S3, S4, R6 and RC).
v. **Class & Object:** [123] defined class as "a piece of code that provides defi-

nition for an object" which includes the behaviour (properties) and methods. It can also be viewed as the "blueprint" of designing (creating) an object. An object belongs to class. We can also call an object as an instance of a class. An object is constructed by instantiating a class. Consider a class call 'Shape' that has property that is 'number of sides' and method 'area'. An instance of the class 'shape' is constructed by adding values to the properties. For example if we put 4 as value for the number of sides, we can name the object created as square. We can use the method 'area' to find the area of the object 'squared' using the property 'number of side'.

vi. **API:** *Application programming interface* is the interactions that occur in the system. For example, interface between one class to another, interface between one class and its properties.

7.2 Background

This section is to provide a review on two things. In the first half of this section, we review some of the existing unified machine learning interface, `scikit-learn` in `Python`, **caret** and `mlr3` in `R`. Then, we review how these toolboxes incorporate distribution estimation into their toolbox.

7.2.1 Background on Existing Unified Interface for Machine Learning

This section is a review on the existing unified interface for machine learning. A mathematical and statistical software is used to solve problems including computation, analysing, modelling and etc. A software may contain packages and toolboxes that serves different purposes. However, the packages and toolboxes may serve the same task but using different functions, commands or interface.

For example, in **R**, the package **randomForest** ([124]) was created to perform random forest for regression (and classification). It contains functions for modelling random forest and other functions to support modelling, plotting, predicting, tuning, setting the tree size and etc. The package **glmnet** ([125]) is used to model ridge regression. Both are used to perform regression (and classification). However, they use different methods and the syntax to train (fit), predict and evaluate the regression task are different.

Example 7.2.1. *Example using* `glmnet`

```
> library(glmnet)
> library(MASS)
> attach(Boston)
> fit.glm = glmnet(x = as.matrix(Boston[,12:13], ncol =
    2), y = Boston[,14])
> newdata = matrix(c(390, 5.4), ncol = 2)
> predict(fit.glm, newx = newdata)
```

Example 7.2.2. *Example using* `randomForest`

```
> library(randomForest)
> library(MASS)
> attach(Boston)
> fit.rf = randomForest(medv ~ lstat + black, data =
   Boston)
> newdata = matrix(c(rnorm(506, mean = 300, sd = 5.3),
   rnorm(506), mean = 5, sd = 3), ncol = 2)
> predict(fit.rf, newdata = newdata)
```

Exmp 7.2.1 and Exmp 7.2.2 show the fitting and predict using `glmnet` and `randomForest` functions, respectively. The type of input for each commands are different. `glmnet` requires `x` and `y` in matrix form but `randomForest` requires input as data.frame. However, notice that training for both packages uses different functions. In contrast, the both packages adopt the same function to predict. This is a confusion that lies within **R** itself because it implements *generic functions*. By using generic function, **R** allows functions to have the same name but different usage and arguments. `predict` act as a dispatcher. When `predict` is called (with arguments), **R** will search along the attributes of the object passed that matches predict function for that object. In our example, `predict` will go through the attributes of `fit.rf` until it finds the function that does predict for it.

The development of a unified ML interface toolbox is to provide a consistent interface to conduct ML algorithm using different functions or learners. In this way, different functions in a software from different packages are collected and follow the same interface for training, predicting and evaluating. Moreover, a unified interface for ML enables a quick and easy implementation, especially for new and non-specialist users. [115] emphasis on the performance, documentation and API consistency for **scikit-learn**. [126] design **caret** by removing the different syntax used in different **R**, packages while **mlr3** was build to standardize the machine learning interface in **R**.

Most unified ML interface have the same structure. Firstly, the data must be defined. Then, the estimator is initialized. Estimator will conduct the fitting and predicting by taking in arguments. Later, the evaluation method is defined that takes in the prediction and evaluate against a dataset. These are the basic steps. Advance machine learning concept such as resampling, tuning, ensemble learning extends the functionalities of the basic steps.

We consider three different ML toolboxes: (1) **scikit-learn**; (2)**mlr3**; (3) **caret**. The design **scikit-learn** and **mlr3** uses the object class relationship where everything is considered as object. We explore the common training, predicting and evaluating methods of a learner fpr the three toolboxes.

Defining Data: Firstly, the dataset used to perform a task needs to be input. How the data is called for the three ML interface is different. For **mlr3**, the data must be defined with its specific task. This means, user must tell what the data is used for, either solving regression problem or classification problem. The data is stored with its properties as an object. The data called is in the ‘data.frame’ format. Example is shown below for the dataset ‘Auto’. The ‘id’ let user to name the task. ‘backend’ is the storage for the dataset and ‘target’ where users specify the label variable.

```
task = TaskRegr(id = "auto", backend = Auto, target = "
   mpg")
```

In **scikit-learn** the dataset used must be of type matrix where the columns are the X and Y variables, i.e features and target variables, respectively.

```
iris = datasets.load_auto()
X, y = auto.data[:, :2], iris.target
```

caret follows similar concept as **scikit-learn**. There is no need to define the dataset used in the learning before hand. The dataset used can be of any type, e.g. data.frame or matrix.

Initialising the Estimators: The next step is to construct the object estimator or learner. **scikit-learn** and **mlr3** uses an OO approach design where the learners are members of a class. For both, once an estimator (learner) is initialized, the object acts as a storing object that attached the parameters to it. The difference is how the estimator is called. Some ML toolbox require user to import manually the learner from other toolbox, this is the case for **scikit-learn** (see Listing 7.1). Whereas some ML toolbox, the methods are imported automatically when importing the ML toolbox itself. This is the case for **mlr3** (see Listing 7.2). Once the estimator is initialized, it calls a function to train the learner on the dataset input. The output after training is also an object model with parameter and data. To train a method in **caret**, the function `train` is called where user require to specify the dataset, label variable and method (see Listing 7.3).

```
from sklearn.linear_model import linear_model
reg = linear_model.LinearRegression
reg.fit(X, y)
```

Listing 7.1: Example for **scikit-learn**

```
learner = lrn("regr.lm")
L1 = learner$train(task)
```

Listing 7.2: Example for **mlr3**

```
fit = train(mpg ˜ . , data = Data, method = "lm")
```

Listing 7.3: Example for **caret**

Prediction: Once the learner is trained, the user can make prediction using the trained model and test (new) data. For **scikit-learn** and **mlr3**, the `predict` function calls the object model together with test dataset. This create another object of type prediction (see Listing and for **scikit-learn** and **mlr3**, respectively). Similar, **caret** have a predict method that calls the fitted model with the test data.

```
y_pred = reg.predict(X)
```

Listing 7.4: Example of Prediction for **scikit-learn**

```
pred = L1$predict(task)
```

Listing 7.5: Example of Prediction for **mlr3**

```
pred = predict(fit, Data)
```

Listing 7.6: Example of Prediction for **caret**

Evaluation: To evaluate the performance of the fitted model, this is done by using the loss function. In **caret** and **scikit-learn**, the method to measure the performance is by calling the prediction and the test data. However, **mlr3** requires to initialize the measure function first and which acts as a storage object. The object prediction will call the measure object.

We show the difference syntax between **scikit-learn**, **mlr3** and **caret** below performing similar task on similar method.

```
mean_squared_error(y_true, y_pred)
```

Listing 7.7: Example of Evaluation for **scikit-learn**

```
msr = msr("mse")
pred$aggregate(msr)
```

Listing 7.8: Example of Evaluation for **mlr3**

```
postResample(pred = pred, obs = Data$mpg)
```

Listing 7.9: Example of Evaluation for **caret**

Example 7.2.3. *Example for* ***scikit-learn***

```
iris = datasets.load_auto()
X, y = auto.data[:, :2], iris.target
from sklearn.linear_model import linear_model
reg = linear_model.LinearRegression
reg.fit(X, y)
y_pred = reg.predict(X)
from sklearn.metrics import mean_squared_error
y_true = []
mean_squared_error(y_true, y_pred)
```

Ex 7.2.3 shows how **scikit-learn** the train, predict and evaluation step using linear regression using the iris dataset. The package that contain the linear regression function is called in the third line. Once the learner is defined (in the fourth line), the `fit` function train the learner `reg`. To evaluate, `mean_squared_error` compares the predicted value with the real values of the label variable.

Example 7.2.4. *Example for **mlr3***

```
library(mlr3)
task = TaskRegr(id = "auto", backend = Auto, target = "
   mpg")
learner = lrn("regr.lm")
L1 = learner$train(task)
pred = L1$predict(task)
msr = msr("mse")
pred$aggregate(msr)
```

Ex 7.2.4 shows example of using **mlr3** to perform prediction using linear regression on the 'Auto' dataset and the target variable is called 'mpg'. `Backend` is stored the data 'Auto'. In **mlr3**, only the packages itself needs to be called. In the second line, the task of the dataset is defined by calling `TaskReg` implying the task is regression. When defining the task, user must tell what is the target variable. The learner is defined before it is trained used for prediction (similar to **scikit-learn**). The method use to evaluates is defined by defining `msr` which called by the `aggregrate` method of the object `pred`.

Example 7.2.5. *Example for **caret***

```
Data = auto.Data
fit = train(mpg ~ ., data = Data, method = "lm")
pred = predict(fit, Data)
obs = mpg
postResample(pred = pred, obs = obs)
```

Ex 7.2.5 shows how to train, predict and evaluation in **caret** in which the evaluation by using the root mean squared error. In **caret**, the learner and the dataset are not required to be initialized beforehand. Rather, the `train` function calls them simultaneously. The evaluation of the trained model is by calling the function **postResample** that compares predicted and true values of the target variable (similiar to **scikit-learn**).

7.2.2 Distribution Estimation in Existing Unified Interface for Machine Learning

This section is a review distribution estimation in ML toolbox. Out of the three toolboxes that we describe in Section 7.2.1, only **scikit-learn** provides an interface for distribution estimation. The distribution estimation is considered as an unsupervised task. The learners included are the kernel based learners with 8 different kernels of choice and the Gaussian mixture. To evaluate the distribution learner, the log-loss is used. The output of prediction is the PDF evaluated at the test data. The example of performing train, predict and evaluate for distribution learner is shown in Listing 7.10.

```
from sklearn.neighbors import KernelDensity
import numpy
kde = KernelDensity(kernel=‘tophat’, bandwidth = 0.5)
kde.fit(X)
kde.score.samples(X)
```

Listing 7.10: Distribution Estimation in **scikit-learn**

weka in **Java** is an open source machine learning interface which also provides an interface for distribution estimation. However, because it uses a different interface, we will not further discuss it here.

7.3 Significance of Unified Interface for Distribution Estimation in R

In this section, we explain the motivation of integrating distribution estimation into the ML toolbox, **mlr3proba**. Toolboxes that include distribution estimation have been discussed in Section 7.2.2. Distribution estimation is fundamental in statistics which is useful for modelling, inferencing, prediction and etc.

In **R**, there exist many packages that are related to probability distribution. We categorized the distribution related packages into three: (1) modelling; (2) estimation; (3) evaluation. Firstly, we categorized **R** packages that have the functionality to output PDF, CDF, quantile function (QF) and random numbers related to a distribution into *modelling* distribution package. An example is the **stats** [112] package which is a core and a build-in package in **R**. It holds many type of distributions. For each distribution, it has the functionality to compute the PDF, CDF, QF and random numbers. Users do not estimate the parameters of a distribution or estimate a distribution using a sample. In contrast, users may have prior knowledge on the parameter(s) of the distribution and the goal is to compute the PDF, CDF, QF or generate random numbers. Example is shown below.

```
> dnorm(1, mean = 0.5, sd  = 1)
[1] 0.3520653
> pnorm(1, mean = 0.5, sd = 1)
[1] 0.6914625
> qnorm(0.1, mean = 0.5, sd = 1)
[1] -0.7815516
> rnorm(1, mean = 0.5, sd = 1)
[1] -0.08308775
```

Estimating distribution packages have the functionality to estimate the parameters and distribution (either PDF or CDF) at a point using the sample data. The **graphic** package holds the function `hist` that estimates a point PDF via histogram estimator using the input (sample) data. *Evaluating* distribution packages involves measuring the performance of distribution. An example of package that able to evaluate distribution is the **scoringRule**.

These three functionalities (estimating, modelling, evaluating) from different packages are not connected and follow different syntax (command). For example, user may have have a dataset and want to find its distribution. The user can use `stats::kde` (stands for `package::function`) which is a function that performs distribution estimation using kernel estimator. The output of this function is not just the PDF, but also the bandwidth, the points where PDF are estimated and etc. The **stats** package does not have a functionality to measure the performance of the this method. Therefore, we need to use **scoringRule** but this package uses different command and input to perform the evaluation. This process makes it tedious for the user.

A summary of packages in **R** that perform the three functions are summarized in Section 7.3.1 and Section 7.3.2 for estimating and modelling, respectively.

7.3.1 Comparisons on Distribution Estimation in Packages & Functions in R

This section is a comparision on some of distribution estimation packages and its functions in **R**. We separate the packages into two groups where kernel based estimators are in Table 7.1 and non-KDE estimators are in Table 7.2. In each table, we provide information of the **R** command use (univariate kernel distribution) for each package, the method used to estimate distribution based on the **R** command, the input of the functions and the type of estimate (output). Although the packages have functions to estimate distribution (the PDF or CDF), some packages have other functions not related to estimating distribution. Some packages contain functions that are able to estimate distribution for more than one dimension or estimate distribution for more than one method.

Here, we provide a brief explanation of the packages and their functions that we included in the **mlr3extraleaners**. A detailed review and description of the packages and their functions has been discussed by [127]. The function for each package we describe in Table 7.1 are for univariate kernel distribution to ensure a consistent comparison. Some of the methods used from the functions have been explained in Chapter 4.

Package	**Properties**
stat [112]	`density(x, bw, kernel)` Estimates univariate point PDF using the kernel estimator with the option of 7 kernel. The sample data is specified by `x` while the bandwidth is specified by `bw`. The default bandwidth is Silverman's rule of thumb as in Eqn (4.2.14). User can specify own bandwidth. The PDF is estimated by dividing the sample data into 512 grid (default).
GenKern [128]	`KernSec(x, xgridsize=100, xbandwidth, range.x)` Estimates univariate PDF at a point for Gaussian kernel. The sample data is defined by `x` and the bandwidth is defined as `xbandwidth`. The default bandwidth is by [68]. The function support variable bandwidth. The PDF is estimated at the point defined by `xgridsize`.
kerdiest [129]	`kde(type_kernel, vec_data, y, bw)` Estimates PDF at a point for univarite kernel estimator with optional of 4 kernels (i.e. Epanechnikov, Gaussian, Quartic and Triweight). The default bandwidth is by [97] as in Eqn (4.2.43). The PDF is estimated at the data points defined by `y` or the function is divided in 100 grids.

ks [130]	`kde(x, h, eval.points)` Estimates the PDF at a point for 1 - 6 dimension of Gaussian kernel estimator. The sample data is defined by `x` and the bandwidth is defined by `h`. The default bandwidth is by [92] but provides option for other bandwidth and user input bandwidth. The PDF is estimated at the points defined by `eval.points`.
sm [131]	`sm.density(x, h, eval.points` Estimates the PDF for 1-3 dimension for Gaussian kernel estimator. The sample data is specified by `x` and the bandwidth is specified by `h`. The default bandwidth is uses 'normal optimal smoothing' by [132]. The function provides functionality to plot the estimated distribution. The PDF is estimated at the points specified by `eval.points`.
TDA [133]	`kde(X, Grid, h, kertype = "Gaussian", weight = 1)` Estimates PDF at a point using kernel estimator with the option of either Gaussian or Epanechnikov kernel. The sample data is defined by `X` and the bandwidth which is defined by `h` has no default option. The PDF is estimated by the data defined by `Grid`. The function can also support variable weigth estimator.
plugdensity	`plugin.density(x, nout, xout)` Estimate PDF at a point using Gaussian kernel estimator using plug-in bandwidth by [134]. The sample data is specified by `x`. The function provide functionality to estimate PDF at the vector specified by `xout`. The PDF can be estimated by number of grid,`nout`

np [135]	`bw = npudensbw(tdat, ckertype)`
	`fitted(npudens(bws=bw, newdata))`
	The package provides different methods to estimate PDF depending on the setting. The method above estimates the PDF for univariate kernel estimator. The method uses two step approach. The bandwidth is first defined with the input sample data `tdat` with the kernel type defined by `ckertype`. The bandwidth is estimated using the data set using the default method by [136]. The PDF is then estimated by calling the defined bandwidth with the data set defined by `newdata`. The package also provided functionality for conditional distribution estimation.

Table 7.1: Table of packages in **R** that estimate distribution using kernel based estimators. The left column is the packages. The right column describe the functions in the features that estimate (univariate) distribution with explaination on features on the functions.

All the functions in Table 7.1 uses different default bandwith. **ks**, **sm** and **TDA** packages support multivariate distribution estimation and **np** supports conditionl distribution estimation.

There are other packages that provide support for non-kernel distribution estimators in **R**. We summarize some of packages and its function in Table 7.2 below. We summarize the input of the function and the method used for estimating distribution.

Package	**Properties**
graphics	`hist(hist(x, breaks, probability)` Estimates the PDF using histogram estimator (refer to Section 2.3). This function also estimates density in terms of frequency. The number of bins is defined by `breaks` with the default method by Sturge rule. By default, the function will plot the histogram. It does not evaluate the PDF at a grid.
TDA [133]	`knnDE(X, Grid, k)` Estimates the PDF using the KNN method (refer to Eqn (2.3.7). The sample data is defined by `X` and the PDFs are estimated at the points specified by `Grid`. This function does not provide any default method for the parameter `k`.

gss [137]	`fit = ssdens( formula)`
	`dssdens(fit, x)`
	`pssdens(fit, q)`
	`qssdens(fit, p)`
	Estimates distribution using the penalized likelihood method. This is a two step approach. An object of `ssdens` is fitted by specifying the data by `formula`. The PDF, CDF and QF can be estimated based by the 'dpq' method of `ssdens` by calling the the fitted object with vectors of points or probabilities.

pendensity [138]	`pendensity(x ~ 1)` Estimates the PDF using the penalized method. The package provide support for univariate distribution estimation and conditional distribution estimation. The function returns an object of class pendensity.

logspline [139]	`fit = logspline(x)`
	`dlogspline(q, fit)`
	`plogspline(q, fit)`
	`qlogspline(p, fit)`
	`rlogspline(n, fit)`
	Estimates distribution using maximum likelihood approach. The method is similar to **gss**. It uses a two-step approach where the model is fitted and return an object of **logspline**. The PDF, CDF, QF an random numbers can be estimated by calling the 'dpqr' function on the fitted model and vectors of d = quantiles, p = probabilities and n = sample size.

Table 7.2: Table of packages in **R** that estimate distribution using other methods than kernel. The left column is the packages. The right column describe the function to estimate distribution with explantion on the features on the functions.

Th package **TDA** have two functions to estimate distribution. The first is in Table 7.1 which uses kernel method and the second is in Table 7.2 using KNN.

7.3.2 Comparison on Distribution Modelling Packages in R

In this section, we compare some of the packages in **R** that allow users to compute the, PDF, CDF, QF and random numbers of a distribution. We first summarize some of the packages **R** that have the functions to perform the tasks of generating PDF, CDF, QF and random numbers of distribution in Table 7.3.

Package	**Properties**
stats [112]	`dX, pX, qX, rX` This corresponds to PDF, CDF, QF and random number generation. X is the name of distribution. For example, for Normal distribution , `dnorm`, `pnorm`, `qnorm`, `rnorm`. The **stats** package contain 17 distributions which includes 4 less common distribution (`pbirth`day, `dsignrank, ptukey` and `dwilcox`) .
distr [140]	`d(X)(),p(X)(),q(X)(),r(X)()` This corresponds to PDF, CDF, QF and random number generation where `X` is the name of the distribution. For example, for Normal distribution, `d(N)(1), p(N)(1), q(N)(0.5), r(N)(1)` where these computes PDF and CDF at 1, quantile function at 0.5 and generate 1 random number. **distr** is the earliest object-oriented distribution package written in S4 language in **R**. In the class design of the package, the parent class is called `Distribution` and have a slot for parameter, `param`, and methods that can simulate, evaluate the PDF, CDF, QF and generate random numbers. The package **Distribution** is inherited by `AbsconDistribution` and `DiscreteDistribution` which are inherited then by distribution class. **distr** has all the distribution implemented in **stats**.
distributions3 [141]	`X(),pdf(X), cdf(X),quantile(X),random(X)` The package is an object-oriented package written in S3language in **R**. The object distribution is first defined and is then used to obtain the PDF, CDF, QF and random numbers.
mistr [142]	`X(),d(X),p(X),q(X),R(X)` This is another object-oriented package written in S3. The package focusses on univariate and composite distribution.
gendist [143]	`dX(),pX(),qX(),rX()` The `X` is the name of the distribution. The package computes the PDF, CDF, QF and random numbers for mixture models, composite models, folded models, skewed symmetric model and arc tan models.

distr6 [114]	`X$new()`, `X$pdf()`, `X$cdf()`, `X$quantile()`, `X$rand()` This is an object-oriented package in R6 format. Apart from computing the normal PDF, CDF, quantile and random numbers of distribution, each distribution able to compute important methods of a distribution (e.g. mean, variance, skewness and etc). Unlike other distribution package in **R**, this package contain 11 kernel methods in addition to 42 distribution.

Table 7.3: Table of packages in **R** that computes PDF, CDF, QF and generate random numbers of a distribution. The left column is is the packages. The right column is a description of features of the packages with function, interface and other information on the packages.

The **stats** package is the basic package in **R** that frequently used by users. There are 4 functions which allow users to compute PDF, CDF, interquantile function and generate random numbers of a particular distribution. This is the usually the basic functionality introduced to new users of **R**. The **distr** package that is an object-oriented version of distribution package in **R**, written in S4 language. The output of the functions called from this package is similar to **stats**. This is done by passing the object of the class distribution to another functions. **distr6** is also an object-oriented distribution, can be seen as an upgrade version of **distr**, written in R6 language. Unlike **distr** and **stats** packages, **distr6** included 11 symmetric kernel methods in addition to its 45 probability distribution. It contains functionality for composite distribution and numerical imputations. Lastly, the package **gendist** support computing the PDF, CDF, QF and random numbers for probability distributions that are useful for actuarial.

7.4 distr6 in R

To incorporate distribution estimation into a unified ML interface, we will use functionalities from **distr6**. Therefore, in this section, we explain the purpose of **distr6** package in the unified ML distibution estimation. For that, we will discuss the design and interface in **distr6**, our contribution in this package and how **distr6** relates to the unified ML for distribution estimation.

distr6 implements object-oriented (OO) probability distribution interface that uses the **R6** language in **R**. This package implements 42 probability distributions and most importantly it implements 11 kernel functions. This is an important feature of **distr6** because nonparametric kernel distributions is the center of this thesis and relates to the contribution of the package. In addition, **distr6** computes the PDF, CDF, QF and simulate random numbers. This functionality is similar to the **stats**, **distr** and other packages discussed in Section 7.3.2. In addition to that, **distr6** has functionalities to design custom distribution, add decorator, use one distribution to create another distribution and others.

Firstly, we recall that a distribution object has the following ([7]):

i. Defining functions, e.g. PDF (Eqn (2.2.1)), CDF (Eqn (2.2.2)) and others.
ii. Properties, e.g. mean, skewness, symmetry.
iii. Types or traits, e.g. continuous, discrete.
iv. In addition, distribution can be used to sample random variable.

Viewing distribution as an object allows a clear class-object design for **distr6** where,

i. Abstract distribution are the classes.
ii. Concrete distribution are objects.
iii. PDF, CDF, QF and other distribution defining functions are the methods of class.

7.4.1 Review on Design of distr6

In this section, we describe the design of **distr6**. We look into the class design and the interface within the package.

Because **distr6** uses OO progamming language, it considers everything that is defined is considered as object which belongs to a class. There are multiple classes in **distr6** but they inherit from the top of the hierarchy which is the abstract class, `Distribution`. The class `Distribution` acts as a container which has properties and methods but not actually implementing them. `Distribution` is inherited by four other abstract classes, `Kernel`, `DistributionDecorators`, `SDistribution` and `DistributionWrapper`. These children classes are more specialized and focussed. They only inherit properties and methods from `Distribution` that are useful for them. Futhermore, the children classes may have their own methods and properties that are not inherited. For example, the class `SDistribution` focusses on parametric distributions. It inherits some of the properties and methods from `Distribution` but also includes methods and properties specifically for the class `SDistribution`. Then, each of the four classes will be inherited by classes that implement concrete distribution. For example, `Kernel` is inherited by `NormalKernel` and includes concrete methods (PDF, CDF and etc) for Normal kernel.

Having explained the class-object design in **distr6**, it is easier to explain the interface within the package for `Kernel` class and between the class `Kernel` and other classes within **distr6**. Below, we explain the interface of creating an object, interface between object and its methods, interface between object and its properties, interface between different classes and interface for creating non-defined distribution.

i. **Interface in creating the object: distr6** has 11 concrete kernel classs, e.g. `NormalKernel`, with concrete methods. To create an object of the class `NormalKernel` it needs to be initialized. This can be done by using the command below.

```
NormalKernel$new()
```

ii. **Interface between the object and methods:** An object of a class has methods. For example, `pdf` is a method that computes the PDF at a point. The object calls the method `pdf` with point (an integer) as shown in the example below.

```
> norm = NormalKernel$new()
> norm$pdf(1)
[1] 0.3989423
```

Here, the function is a kernel function, $K : \mathbb{R} \to \mathbb{R}^{+}$. Other methods maybe `cdf` and `qf` which computes the CDF and quantile function at a point.

iii. **Interface between object and properties:** An object also has properties or (and) parameters. Interface between the object and its properties allows users to get or change the properties. To get the parameter, the user can use the command

```
$getParameterValue()
```

or to set the parameter

```
$setParameterValue()
```

A more clear example is shown below where an object of class Normal distribution is created. From the object, user can access to the property of the object, in the example below we show the property is the median. Then, user can obtain the value of the parameter `mean` and later change that value.

```
> norm = Normal$new(mean = 1, sd = 1)
> norm$median()
[1] 0.3989423
> norm$getParameterValue("mean")
[1] 1
> norm$setParameterValue("mean" = 2)
> norm
Norm(mean = 2, var = 1, sd = 1, prec = 1)
```

Currently, the class `Kernel` does not have any parameter. Hence, we provide an example that is not within the `Kernel` class.

iv. **Interface between classes:** This interface allows the interaction between different classes. For example the interaction between the class `SDistribution` and the class `DistributionDecorator`. A decorator is useful to functionality to an object. This interface allows object of any concrete class that inherits from `SDistribution` to implement methods in the `DistributionDecorator`. The concrete class has only a number of methods. This interface extends the methods of a concrete class to compute methods not usually used, such as n-th moment or p-norms. It is important to note that implementation of this methods are computed numerically. The `DecoratorDistribution` does not actually have methods such as n-th moment or p-norm for each concrete distribution or kernel class.

v. **Interface for non-defined distribution:** An advantage of **distr6** is the ability for user to define own's distribution not implemented in **distr6**. The construction uses the constructor from the class `Distribution` directly.

7.4.2 Contribution to distr6

In this section, we highlight our contribution in **distr6** which is more focussed on the concrete kernels classes that inherits the abstract class `Kernel`. The contribution is an extension of Chapter 5. Currently, the package **distr6** only contain methods to compute the PDF, CDF and QF of the kernel functions. From Chapter 5, a closed-form expressions of the L2-norm of PDF and L2-norm of CDF of these kernels were derived (see Appendix B.1) which is useful to compute the proper loss functions for distribution estimation. Therefore, we incorporate public methods consisting of the L2-norm of PDF and L2-norm of CDF for all concrete kernel classes except for the class `Normalkernel` and `Sigmoid`. This addition allows user to compute the

PSL (Eqn 3.3.11) and IBL (Eqn (3.3.13)) for distribution estimation.

There are a few reasons why we add the public methods consist of the L2-norm of PDF and L2-norm of CDF to concrete kernel classes. Firstly, although the L2-norm methods for distribution are currently available to be computed via decorators using *ExoticStatistics*, these functions are not available all for distribution or kernels. Secondly, constructing a new `Decorator` object is not allowed unless it has at least three methods. Finally, decorators compute methods by imputing functions (using other functions and not directly computing itself) and computation of the L2-norm is done numerically.

In implementing the analytical expression of the L2-norm of PDF and CDF, we focus on the followings:

i. **The need to follow design of distr6:** In accordance to **distr6** design principal, it should separate the implementation of numerical and analytical results. In which, analytical should be via the distribution itself and numerical is imputed using decorators. Therefore, the analytical methods of L2-norm of PDF and CDF are implement as part of public methods of `XKernel` class (i.e. X is the name of kernel, e.g. NormalKernel).

ii. **Precision results:** We want to ensure that the L2-norm of PDF and CDF computed is precise. Which is the reason to implement the analytic expression.

iii. **Easy to implement:** By implementing the L2-norm of PDF and CDF methods for each kernel allows users to directly compute them without the need to go through the steps of decorating the object kernel.

It is important to note that, not all kernel classes have methods to compute L2-norm PDF and CDF. The class `NormalKernel` and `Sigmoid` do not have methods for L2-norm of CDF because the analytical expression of both methods do not have a closed-form. Furthermore, all concrete class kernels do not allow users to define the parameters, which is for kernel the parameter is the bandwidth. Therefore, to compute the PDF, CDF and other methods of distribution for kernels, some manipulations need to be done outside. The interface between the object kernel and the methods L2-norm of PDF and CDF are shown below.

i. **L2-norm of PDF** as in Eqn (5.4.2)

```
Xkernel$pdfSquared2Norm(x, upper)
```

ii. **L2-norm of CDF** as in Eqn (5.4.3)

```
    Xkernel$cdfSquared2Norm(x, upper)
```

X is the name of kernel, (e.g. NormalKernel$pdfSquared2Norm). For Xkernel$pdfSquared2Norm(), the inputs x (centre point) and upper (upper boundary) by default are x = 0 and upper = ∞. This is similar for Xkernel$cdfSquared

7.4.3 Examples

In this section, we provide some example of usage to compute the L2-norm for PDF and CDF.

Example for 'vanilla' kernel

Below is an example for evaluating the PDF, CDF, L2-norm of PDF and CDF for vanilla kernel function as in Def 5.4.1. In this example, we compute the PDF of a uniform kernel, the CDF as in Eqn (5.4.1) where $a = 0.3$, L2-norm of the uniform PDF as in Eqn (5.4.2) where $a = 1, c = 0.1$ and L2-norm of CDF as in Eqn (5.4.3) where $a = 0, c = 0.1$ and L2-norm of CCDF as in Eqn (5.4.12) with $a = 0, c = -0.1$. Then, the losses are computed using log-loss, PSL and IBL as in Def 5.4.2.

```
> uniform = UniformKernel$new()
> # Compute the PDF
> uniform$pdf(0.1)
[1] 0.5
>  #Compute the CDF
> uniform$cdf(0.3)
[1] 0.65
> # Compute the L2-norm of the PDF of a kernel
> uniform$pdfSquared2Norm(x = 0.1, upper = 1)
[1] 0.475
> # Compute the L2-norm of the CDF of the kernel
> uniform$cdfSquared2Norm(x= 0.1, upper = 0)
[1] 0.070875
> # Compute the L2-norm of the CCDF of the kernel
> uniform$cdfSquared2Norm(x= -0.1, upper = 0)
[1] 0.09583333
> # Compute the log-loss of the PDF of a kernel
> -log(uniform$pdf(0.1) )
[1] 0.6931472
```

```
> # Compute the PSL of the kernel
> -2*uniform$pdf(0.1) + uniform$pdfSquared2Norm(x =
   0.1)
[1] -0.525
> # Compute the IBL of the kernel
> uniform$cdfSquared2Norm(x= 0.1, upper = 0) +
   uniform$cdfSquared2Norm(x= -0.1, upper = 0)
[1] 0.1667083
```

Examples for uniform mixture kernel

Below is an example for evaluating the PDF, CDF, L2-norm of PDF and CDF for uniform homogeneous kernel mixtures methods as in Prop 5.4.1. The outputs are objects. These objects are then used to compute the log-loss, PSL and IBL of uniform homogeneous kernel mixture as in Section 5.4.2.1.

i. **PDF and Log-loss:** To compute the mixture PDF at a point a for uniform kernel is shown below where $h = 0.4$, the point to estimate the PDF is $x = 0.3$ and the observations x_i where $i = 1, 2$ from a uniform weight homogeneous kernel is $0.5, 0.7$. Then, the log-loss for the object PDF can be computed.

```
> sample = c(0.5, 0.7)
> h = 0.4
> x1 = 0.3
> dx = sapply(sample, function(x) (x - x1) /  h)
> uniform = UniformKernel$new()
> uniform.pdf = mean(uniform$pdf(dx)/h)
> -log(uniform.pdf)
[1] -0.2231436
```

ii. **CDF:** To compute the mixture CDF at a point a for uniform kernel is shown below using Algorithm 12. We let $h = 0.4$, the point to estimate the CDF is $a = 0.3$ and the observations x_i where $i = 1, 2$ from a uniform weight homogeneous kernel is $0.5, 0.7$.

```
> sample = c(0.5, 0.7)
> h = 0.4
> a = 0.3
> d = sapply(sample, function(x) (x - a) / h )
> uniform.cdf = mean(uniform$cdf(d))
> uniform.cdf
[1] 0.875
```

iii. **L2-norm PDF and PSL:** To compute the L2-norm of mixture PDF and the PSL for uniform kernel as shown in Algorithm 10. The PSL is evaluated at $x = 0.3$. Let $h = 0.4$, $c = 0$ and the observations x_i where $i = 1, 2$ from a uniform weight homogeneous kernel is $0.5, 0.7$.

```
> sample = c(0.5, 0.7)
> h = 0.4
> x1 = 0.3
> dx = sapply(sample, function(x) (x - x1) /  h)
> uniform = UniformKernel$new()
> uniform.pdf = uniform$pdf(dx)/h
> #L2-norm pdf for c = 0
> d = sapply(sample, function(x, y) (x - y)/h, y =
   sample)
> uniform.pdf2norm = sum(uniform$pdfSquared2Norm(x
   = d, upper = Inf)*(1/2)^2 * 1/h)
> uniform.pdf2norm
[1] 1.09375
> #compute the PSL
> mean(-2 * uniform.pdf + uniform.pdf2norm)
[1] -1.40625
```

iv. **L2-norm CDF, L2-norm CCDF and IBL:** To compute the L2-norm of CDF mixture, L2-norm of CCDF mixture and IBL for Uniform kernel as shown in Algorithm 11. The IBL is evaluated at $x = 0.3$ We let $h = 0.4$, $c = 0$, and the observations x_i where $i = 1, 2$ from a uniform weight homogeneous kernel is $0.5, 0.7$.

```
> sample = c(0.5, 0.7)
> h = 0.4
> x1 = 0.3
> #L2-norm cdf for c = 0, upper = x1
> da = sapply(sample, function (x) (x1 -x)/h)
> dx = sapply(sample, function (x, y) (x - y) / h,
   y = sample)
> #compute the L2-norm of cdf
> uniform.cdf2norm = (1/2)^2 * h * sum(
   uniform$cdfSquared2Norm(x = da, upper = dx))
> #compute the L2-norm of ccdf
> uniform.ccdf2norm = (1/2)^2 * h * sum(
   uniform$cdfSquared2Norm(x = - da, upper = -dx))
> #compute the IBL for Uniform Mixture
> uniform.cdf2norm  + uniform.ccdf2norm
[1] 0.08854167
```

7.4.4 Purpose of distr6 for Unified ML for Distribution Estimation

In this section, we clarify the significance of **distr6** for the unified ML interface for distribution estimation. In distribution estimation, there are two outputs: (1) PDF and (or) CDF; (2) `distr` object. The `distr` object is a customize object created from the function `decorator` of **distr6**. This means, that any distribution estimation learner will predict an object of **distr6**. For example is a histogram distribution which estimate the PDF at a point is not defined in **distr6**. However, a custom distribution of **distr6** object can be created for estimated histogram shown below. See Section 7.5.4 for further detail.

```
distr = distr6::Distribution$new(
      name = "Histogram Estimator",
      short_name = "Histogram",
      pdf = pdf, cdf = cdf,
      type = set6::Reals$new(),
      support = set6::Interval$new(min(fit$breaks), max
         (fit$breaks)))
```

In the above code, we use the function `Distribution` from **distr6**. The name and short name of the object are specified. For the estimated histogram, the point PDF and CDF are computed. `type` is to inform users that the traits of PDF and CDF.

7.5 Design & Computation of Unified ML Interface for Distribution Estimation

In this section, we explain the unified interface for distribution estimation. [113] discussed the used of **mlr3proba** for survival analysis. In this section, we discuss the interface of **mlr3proba** for distribution estimation. This interface uses multiple packages that already exist in **R**. It connects **mlr3**, **mlr3proba**, **distr6**, **mlr3extralearners** and other distribution *estimation* packages from Table 7.1 and Table 7.2 together to provide an easy implementation for users to conduct machine learning steps. In this section, we try to explain the following:

i. Why is the development uses **mlr3**?
ii. What is the design of the interface?
iii. How can users implement this?

The outline of this section is as follows. We begin by briefly explaining the connecting between **mlr3**, **mlr3proba**, and **mlr3extralearners** for this interface. Then, we describe how the class design, interface and explain what is being computed.

7.5.1 Interface for Distribution Estimation

In this section, we discuss the design of distribution estimation interface and connecting **mlr3**, **mlr3proba**, **mlr3extralearners**, **distr6** and other packages in Section 7.3.1 and Section 7.3.2. This interface focusses on the nonparametric univariate distribution estimation.

Using mlr3

Currently, there exist four ML interface in **R**, i.e. **mlr3**, **tinymodels**, **mlr** and **caret**. **mlr3** is an object-oriented approached and has the advantage to reuse the object. It has a very solid foundation and only focusses on the core functionality. Other functionalities such as tuning and benchmarking are accessablle via **mlr3**'s 'child' pacakage, **mlr3tuning**, **mlrbenchmark** and others. **caret** on the other hand (as its name) is more focussing in regression and classification task. This itself restrict us from implementing an ML interface for distribution estimation. **mlr** is also an OO package for machine learning developed by the same team as **mlr3**. However, **mlr** is overloaded with all tasks falls into the package itself.

mlr3 is a core package that provides a unified interface for regression and classification tasks. In addition, **mlr3** has an *ecosystem*. We can define it as different packages that extend the **mlr3** functionality. Some of the packages provide support to extend the basic train, predict and evaluate in **mlr3**. For example, the package **mlr3tuning** provides support for learners of **mlr3** to perform hyperparameter tuning and nested resampling. Then, there is the **mlr3pipeline** that enable the creation of ensembler learner from the basic learners in **mlr3**. Within the ecosystem, there are packages that provide different task for machine learning other than regression and classification, for example **mlr3cluster** and **mlr3proba**.

mlr3proba supports the probabilistic supervised learning. Currently, it has the interface for survival analysis. Since distribution is a probabilistic learning, we integrate into **mlr3proba** a unified interface for distribution estimation.

7.5.2 Design of mlr3proba and Unified Interface for Distribution Estimation

In this section, we discuss the class design for distribution estimation. Although our main focus are integrating the task, learners, prediction and measure classes to support unified ML for distribution estimation, it is important to connect the relationship between classes in **mlr3**.

Task: The class `Task` sits on the topmost of the hierarchy. It is an abstract class that does not implement any methods but it act as mold. It defines what properties or methods an object `Task` could (should) have. `Task` is inherited by other children classes, `TaskSupervised` and `TaskUnsupervised`, which are also abstract. Following that are grandchildren classses which are concrete classes with methods and properties, `TaskRegr`, `TaskClassif`, `TaskSurv`, `TaskClust` and of course `TaskDens`. The UML diagram of the classes for Task are shown in Figure 7.1.

The object of any concrete class `Task` holds the meta-information of the data such as the type of task, the target variable (i.e. for supervised task) and others.

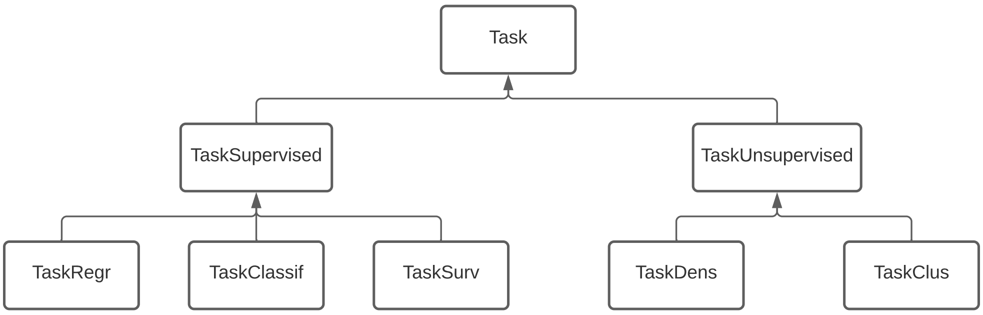


Figure 7.1: Figure of inheritence of the class `TaskDens`.

Learner: Similar to the design structure of task, the abstract class `Learner` sits on the top and is inherited by abstract classes `LearnerRegr`, `LearnerClassif`, `LearnerClust`, `LearnerSurv` and `LearnerDens`. The children classes have defined properties and methods. For `LearnerDens`, it is inherited by concrete learner classes which have methods train and predict. For example, `LearnerDensKDE`, `LearnerDensHistogram` and others. **mlr3proba** does not contain all distribution learners, only the two mentioned. Other distribution learners are collected in one of the **mlr3** ecosystem, which is **mlr3extralearners**. Table 7.4 shows the separation of ecosystem for distribution learner. The other learners in **mlr3extralearners** collects different distribution learners from different packages in **R**. These learners are from Table 7.1 and 7.2.

Learner	
LearnersDens(mlr3proba)	**LearnerDens(mlr3extralearners)**
LearnerDenHistogram	LearnerDensKDEks
LeanerDensKDE	LearnerDenKDEkd
	LearnerDenMixed
	LearnerDenNonparametric
	LearnerDenLocfit
	LearnerDenLosgspline
	LearnerDenPenalized
	LearnerDenPlugin
	LearnerDenSpline

Table 7.4: Table of learners in **mlr3proba** and **mlr3extralearners**.

The object `LearnerDens` have two functions. First is to store information and bind parameters and second is to output an object model after training. This will further discuss in Section 7.5.3.

Prediction: The design for class structure for prediction follows the similar structure to `Task`, in which the abstract class `Prediction` is inherited by abstract classes `PredictionRegr`, `PredictionClassif`, `PredictionClust`, `PredictionSurv` and `PredictionDens`. The output of `PredictionDens` is an object of type prediction which stores the prediction after learning. The object `PredictionDens` is a table that contains information which includes the id of the data used for prediction, the PDF and the type of distribution.

Measure: The class structure for measure follows the same structure as the design for learner. The class `Measure` is an abstract inherited by abstract class `MeasureDens`, `MeasureRegr`, `MeasureClassif`, `MeasureClust` and `MeasureSurv`. These classes are inherited by concrete class with methods. The current version of **mlr3proba**, the class `MeasureDens` inherited by `MeasureDensLogloss` that have methods to compute the empirical log-loss (Eqn 3.3.10). The object `MeasureDensLogloss` compute the empirical generalization log-loss.

7.5.3 Interface for unified ML interface for distribution estimation in mlr3proba

In this section, we explain the interface for unified ML interface for distribution estimation. We look into different interface, i.e. interface for contruction of objects, interface between classes, interface between a class and its methods or properties and others.

i. **Interface of constructing objects:** The object for task, learner and measure should be instantiated. First, consider constructing an object of class `TaskDens`. This can be done by using the command,

```
task = TaskDens$new(id, backend)
```

The constructor takes in 2 important arguments, the 'id' and the 'backend' which are the name user give for the task (a type string) and the data that is used to learn distribution estimation, respectively. The object that is created act as a storage of meta-information. Constructing the object of any concrete distribution learner is by,

```
lrn = lrn(id, param_set)
```

The ‘id’ is a type of string that defines the concrete learner, for example,

```
lrn = lrn("dens.kde", bandwidth = 0.1, kernel = "Norm")
```

Above shows the how to construct the learner “dens.kde”. This object act as a storage with the parameters bind to it. Similarly, the object of measure for distribution estimation should be initialized by the command,

```
msr = msr("dens.logloss")
```

The defined object of measure act as a storage and has yet to implement its method.

ii. **Interface between class task and class learner:** This interface connects the object task and learner by calling the `train` function of the object learner together with the task,

```
L1 = lrn$train(task)
```

The object learner is trained using the task and output an object model.

iii. **Interface between class learner and prediction:** This interface connects the object model that has been learned (trained) with the prediction object. Once the object model is learned, calling the `predict` method of the object learner and `Task` to create an object of class `PredictionDens`,

```
L1$predict(task)
```

In the above, the prediction is being done on the same task data.

iv. **Interface between class prediction and class measure:** Once the object prediction of class `PredictionDens`, it calls the score function and the instantiated object measure,

```
prediction$score(msr)
```

The score is computed using the prediction object.

7.5.4 Return Type

In this section, we will describe the return types of the prediction. In this section, we explain how **mlr3proba** and **distr6** are connected.

In distribution estimation, we aim to return an object that is reproducible. Rather than returning the predicted PDF or CDF, we want to ensure that users are able to obtain the distribution of the PDF. Therefore, for distribution estimation, we can obtain multiple outputs. We recall in Chapter 2, that a distribution is an object defined by functions PDF, CDF and etc. The functions estimate the PDF, CDF and QF at point. In **mlr3proba** (also for distribution learners in **mlr3extralearners**), we standardized so that the output of the prediction method are: (1) `pdf` estimated at the test data; (2) **distr** which is a distribution object implement via **distr6**. Some learners will have `cdf` estimated at the test data. This standardization is important because each distribution learners from different package have different output. This standardization is also advantages to compare different learners.

In Section 7.4.4, we described briefly the purpose of **distr6** in the unified ML interface for distribution. Using the functionality of **distr6** to create a custom distribution allow users to use the 'predicted' distribution for other purpose. For example, when using kernel learner, other than obtaining the predicted PDF at the test data of the kernel learner, the predicted `distr` object allows users to compute CDF, variance and L2-norms of the CDF and PDF. An example of the interface between **mlr3proba** and **distr6** is shown below.

```
> data = data.frame("A" = as.numeric(rnorm(30)))
> task = TaskDens$new(id = "a", data$A)
> train_set = sample(task$nrow, 0.7 * task$nrow)
> test_set = setdiff(seq_len(task$nrow), train_set)
> a = task$data(train_set)
> L1 = lrn("dens.kde", kernel = "Epan")
> learner = L1$train(task, train_set)
> learner$model$bandwidth
[1] 0.5809963
> prediction = L1$predict(task, test_set)
> prediction
<PredictionDens> for 9 observations:
    row_id          pdf              distr
         5 0.20444922 <Distribution[38]>
         6 0.18540074 <Distribution[38]>
         7 0.41451000 <Distribution[38]>
---
        21 0.06851355 <Distribution[38]>
        22 0.04405488 <Distribution[38]>
        23 0.22770858 <Distribution[38]>
>kernel = get(as.character(subset(
          distr6::listKernels(),
          ShortName == unlist(
             prediction$distr$parameters("kernel")
             [[2]]),
          ClassName)))$new()
> kernel
Epan()
> kernel$pdfSquared2Norm(x = 0.1, upper = 0.5)
[1] 0.516117
> kernel$cdf(0)
[1] 0.5
> kernel$variance()
[1] 0.2
```

7.5.5 Examples

In this section, we provide some examples for distribution estimation interface for three different cases: (1) the vanilla train, test and evaluate; (2) optimization of bandwidth using KDE estimators; (3) benchmarking experiment in comparing different learners with benchmarking experiment included using nested cross-validation.

i. Firstly, we provide the vanilla case for training, prediction and evaluation using kernel estimator and variable 'mpg' of the dataset 'mtcars' from the UCI database ([111]). In this example, we split the data into training and test sets. We use the default kernel (Epanechnikov) and default bandwidth.

```
> #initialize the task
> Task1 = TaskDens$new(id = "mpg", backend = datasets::
   mtcars$mpg)
> #split into training and test
> train_set = sample(Task1$nrow, 0.8 * Task1$nrow)
> test_set = setdiff(seq_len(Task1$nrow), train_set)
> #initialize learner
> L1 = lrn("dens.kde")
> #train/fit the learner on the train set
> learner = L1$train(Task1, train_set)
> learner$model$bandwidth
[1] 2.663521
> #predict on the test set
> prediction = L1$predict(Task1, test_set)
> prediction
```

```
<PredictionDens> for 7 observations:
 row_id         pdf              distr
      6 0.050941218 <Distribution[38]>
     10 0.054495458 <Distribution[38]>
     14 0.073300047 <Distribution[38]>
     20 0.007691089 <Distribution[38]>
     22 0.074871814 <Distribution[38]>
     30 0.058147036 <Distribution[38]>
     32 0.064411327 <Distribution[38]>
> #initialize the measure
> m = msr("dens.logloss")
> #evaluate the prediction
> prediction$score(m)
dens.logloss
    3.078119
```

ii. Below, we show the example of tuning the parameters for distribution estimation via **mlr3tuning**. In this example, we tune the bandwidth using the same Epanechnikov kernel estimator and 'mtcars' dataset. We use the same 'task', learner 'L1' and measure 'm' from previous example.

```
> library(mlr3tuning)
> # initialize the resampling method
> resample = rsmp("holdout")
> # define the search space for bandwidth
> ps = ParamSet$new(
+   params = list(ParamDbl$new("bandwidth", lower =
   0.001, upper = 1)))
> trm = trm("evals", n_evals = 5)
> # define which tuning method to use
> tuner = tnr("grid_search")
> at = AutoTuner$new(
+   learner = L1,
+   resampling = resample,
+   measure = m,
+   search_space = ps,
+   terminator = trm,
+   tuner = tuner)
```

```
> #execute
> at$train(Task1)
> at$learner
<LearnerDensKDE:dens.kde>
* Model: list
* Parameters: kernel=Epan, bandwidth=0.778
* Packages: distr6
* Predict Type: pdf
* Feature types: integer, numeric
* Properties: missings
```

iii. Below, we provide example of performing benchmarking experiment on two kernel learners and two datasets. We continue to use ‘Task1’, ‘L1’ and ‘m’ for the task, learner and measure. In the example below, we initialized another task and learner.

```
> library(MASS)
> attach(Boston)
> Task2 = TaskDens$new(id = "boston", backend =
   Boston$medv)
> L2 = lrn("dens.kde", kernel = "Norm", bandwidth =
   0.001)
> #design of the experiment
> design = benchmark_grid(tasks = c(Task1, Task2),
+                          learners = c(L1, L2),
+                          resamplings = resample)
> bmr = benchmark(design)
INFO  [17:43:23.174] Benchmark with 4 resampling
   iterations
INFO  [17:43:23.182] Applying learner ’dens.kde’ on
   task ’boston’ (iter 1/1)
INFO  [17:43:23.449] Applying learner ’dens.kde’ on
   task ’mpg’ (iter 1/1)
INFO  [17:43:23.470] Applying learner ’dens.kde’ on
   task ’boston’ (iter 1/1)
INFO  [17:43:23.717] Applying learner ’dens.kde’ on
   task ’mpg’ (iter 1/1)
INFO  [17:43:23.739] Finished benchmark
```

```
> # compute the empriical loss
> rr = benchmark(design)$aggregate(msr("dens.logloss"))
> rr
   nr      resample_result task_id learner_id
      resampling_id iters dens.logloss
1:  1 <ResampleResult[21]>     mpg   dens.kde
   holdout     1     3.202921
2:  2 <ResampleResult[21]>     mpg   dens.kde
   holdout     1    24.316121
3:  3 <ResampleResult[21]>  boston   dens.kde
   holdout     1     3.448817
4:  4 <ResampleResult[21]>  boston   dens.kde
   holdout     1    11.541058
```

7.6 Conclusion

In this chapter, we provide a platform of unified ML interface for distribution estimation. This unified interface enable the train, predict and evaluate steps for distribution estimation. Since **mlr3proba** is within the **mlr3** ecosystem, user can use the extension of ML algorithms for distribution estimation such as tuning hyperparameter tuning, benchmarking and etc. Overall, the integration of distribution estimation into **mlr3proba** allows quick and easy implementation for users. However, the functionality provided by this interface is still limited, for example the score function. With the integrated functionalities of L2-norm of PDF and CDF in **distr6**, we can add the measures PSL and IBl into unified interface (for kernel learners). As this is still in the working phase as we need to consider standardization and computation time. This will be a part of future work and is explained further in Chapter 9.

Chapter 8

Benchmarking Experiment

8.1 Introduction

The objective of this chapter is to compare and investigate the performance of multiple distribution learners on multiple datasets by performing a benchmarking experiment. The performance of the distribution learners are measured using log-loss, PSL and IBL. From this experiment, the compared learners are ranked according to the loss functions.

The learners that are being compared consist of nonparametric distribution learners. The learners are grouped into two categories: (1) Kernel based distribution; (2) Non-kernel based distribution. The reason is because on top of evaluating the performance of all distribution estimator using the log-loss, we evaluate the kernel-based distribution learners using PSL and IBL.

In total, the experiment will consist of 29 distribution learners and benchmark on 54 datasets. The datasets are obtained from the UCI ([111]) database. For evaluation purpose, each dataset will be split by 3-fold cross-validation. In addition, for learners that require tuning, each training fold will be further split by 3-fold cross-validation (i.e. using nested resampling method). The mean out-of-sample empirical expected generalization loss (we use 'mean empirical loss' for short) over 3-fold of each dataset and each learner is computed. Then, the average of mean empirical loss over all datasets for each learner is reported. The reported results are then used for further analysis using Friedman test.

This chapter is organized as follows. First, we describe the background and general framework of benchmarking experiment. Second, we look into some literature on experiments that compare distribution learners. Later, we explain the benchmarking framework to fit our benchmarking experiment for distribution estimation in which we specify the datasets, learners, loss functions and resampling methods used in the experiment. Lastly, we discuss the result of the experiment.

8.1.1 Background of Benchmarking Experiment

In this section, we review the theoretical background on benchmarking experiment which is based on [144] to gain a better understanding on benchmarking experiment objectives.

Benchmarking experiment is an empirical investigation that is used to compare the performance of multiple learners of a specific learning task with respect to a certain measurement (loss function). The need to conduct benchmarking experiment arise from different problem. Some examples of problems that use benchmarking experiment addressed by [145].

i. Sensitivity analysis - investigating the affect of different structure of the data generating process on an algorithm.

ii. Algorithm comparison - comparing the performance of a set algorithms $\hat{f}_k$ where $k = 1, \ldots, K$ on a single or multiple data generating process.

Some of the goals for performing benchmarking experiment are:

i. To compare the performance of learners of specific task with respect to a measure.

ii. To compare the performance of new learner or algorithm with existing learners

iii. To compare the performance of learners of a specific task based on groups.

8.1.2 General Framework for Benchmarking Experiment

This section is an overview of a general set-up for benchmarking experiment. This set up has been discussed by [144] and [145]. A benchmarking experiment is divided into three stages ([145]): (1) design; (2) execution; (3) analysis. We will explain the three stages below.

Design

i. First, consider $B \geq 1$ datasets,

$$\mathcal{D}^1 =\{(x_1^b, y_1^b), \ldots, (x_N^b, y_N^b)\}$$

where $b = 1, \dots, B$.

ii. Second, the task for the datasets is specified, (e.g. classification, regression or distribution estimation).

iii. Third, the learners (or algorithms) to perform the task are defined. Learners can be of the same estimator with different parameter input or even estimator of different kind (e.g. to perform regression will have the choice of linear regression, tree regression, etc.). We let the learners be $\hat{f}^k$ where $k = 1, \dots, K$ (i.e. K different learners).

iv. Then, the resampling method is defined to ensure that fitting and testing is done using different datasets. This is important so that function is fitted in the training set and evaluated using the test test. Resampling method may be either hold-out resampling or cross-validation. Let $\mathcal{T}^b = \{(x_1^b, y_1^b), \dots, (x_n^b, y_n^b)\}$ be the training set of dataset $\mathcal{D}^b$ while $\mathcal{T}^{*b} = \{(x_1^{*b}, y_1^{*b}), \dots, (x_m^{*b}, y_m^{*b})\}$ be the test set, where $m + n = N$.

v. Lastly, the methods to evaluate the performance of the learners for a specific task. This is where loss function is specified, for example for regression task the loss function is $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The loss function is used to compute out-of-sample empirical loss function, $\mathbb{E}_{emp}[\mathcal{L}(\hat{f}^k, \mathcal{T}^{*b})]$ which is to measure the performance of the learner $\hat{f}^k$ on the test dataset, $\mathcal{T}^{*b}$.

Execution

Once the design of the experiment has been set up, it is ready to be executed. At this stage, each learning algorithm $\hat{f}^k$ is fitted on each training dataset $\mathcal{T}^b$. Each model is conditioned on $\mathcal{T}^b$. The fitted model includes parameter and dataset. Once the learner is fitted, the performance of each learner on each dataset is evaluated on the test dataset $\mathcal{T}^{*b}$ using the loss function $\mathcal{L}$ by computing the out-of-sample empirical generalization loss.

Analysis

The benchmarking experiment will output the mean out-of-sample empirical generalization loss for each dataset and each learner. Multiple ways may be used to analyse the result of the benchmarking experiment depending on task and objective of the experiment. The learners can be compared by the averaging the mean out-of-sample empirical generalization loss over all datasets (which we refer to mean loss). The average rank of the learners can be found. First, the rank of the mean out-of-sample empirical generalization loss of each learners are found for each dataset. Then, the rank is average over all datasets for each learner. Another way to analyse is by running hypothesis tests on the mean out-of-sample empirical generalization loss. Other analysis methods include relative efficiency and by using graphic.

8.1.3 Literature on Simulation & Benchmarking Experiment

Machine learning is a fast growing field of study that is constantly developing. New methods to achieve a specific task is being proposed and benchmarking experiment is a good tool to compare all learners for a specific task. Literatures on benchmarking experiment for supervised task, includes comparing classification learners by [146], [147], comparing regression task by [148] and even time series classification task by [149].

Over the years, new methods for distribution estimation have been proposed, including new estimators, algorithms, parameter selection and etc. Simulation studies is used to compare new methods with the existing or baseline method. However, comparison of distribution learners are done differently.

8.1.4 Exeriments on bandwidth selection Methods

Table 8.1 shows some of the literatures that compares the performance of distribution learners obtained by different bandwidth selection method.

Paper	Data	Learners	Findings
[67]	4 simulated datasets	$\hat{h}_{BCV}$, $\hat{h}_{LSCV}$, $\hat{h}_{PM}$, $\hat{h}_{OS}$([95], [67])	$\hat{h}_{PM}$ performed better for standard normal and Gaussian mixture with different mean while h_{LSCV} is good for skewed distribution. h_{BCV} and h_{LSCV} has problems with minimizer outside the range.
[68]	4 simulated datasets	$\hat{h}_{PM}$, $\hat{h}_{SJ}$, $\hat{h}_{SJ1}$, $\hat{h}_{SJ2}$	Overall $\hat{h}_{SJ1}$ performed better for standard Normal distribution datasets, Gaussian mixtures with different mean and Gaussian mixture with different variance. $\hat{h}_{PM}$ performs well for normally distributed dataset.
[69]	3 simulated datasets with different sizes	$\hat{h}_{HO}$, $\hat{h}_{ROT}$, $\hat{h}_{PM}$	$\hat{h}_{HO}$ has a better performance when the simulated data is from standard Normal distribution and mixture of Normals with different variance.

[91]	15 simulated datasets as in [99]	$\hat{h}_{ROT}$, $\hat{h}_{LSCV}$, $\hat{h}_{BCV}$, $\hat{h}_{SJ}$	$\hat{h}_{ROT}$ has high mean and low variance $\hat{h}_{LSCV}$ is centred correctly but too spread out $\hat{h}_{BCV}$ has erratic distribution $\hat{h}_{SJ}$ performance is between h_{rot} and h_{LSCV}
[97]	15 simulated datasets as in [99]	ECDF, $\hat{h}_{ROT}$, $\hat{h}_{PB}$-one step and $\hat{h}_{PB}$-two step	$\hat{h}_{PB}$-two step has a better performance for separated bimodal distribution dataset.
[150]	1 real world dataset	$\hat{h}_{LSCV}$, $\hat{h}_{SJ}$	$\hat{h}_{SJ}$ is recommended for its overall performance
[70]	9 simulated mixture datasets from [99] of different size each replicated 10000 times	$\hat{h}_{Hk}$ with $k = 1, \ldots, 8$, $\hat{h}_{HO}$	$\hat{h}_{HO}$ and k-step (1 - 8) perform similar for Normal distributed, skewed, strongly skewed, bimodal and asymmetric bimodal datasets. For kurtotic dataset, datasets with outlier and separated bimodal dataset, $\hat{h}_{Hk}$ with $k \geq 4$ have better performance.
[151]	4 simulated datasets from [99] of different size each repeated 1000 times	$\hat{h}_{LSCV}$, $\hat{h}_{BCV}$, $\hat{h}_{SJ}$, $\hat{h}_{SJ1}$, $\hat{h}_{CONT}$ ([152])	MISE decreases as number of data points increases. Learners are ranked from best to worst: $\hat{h}_{CONT}$ $\hat{h}_{SJ}$ & $\hat{h}_{SJ1}$, $\hat{h}_{LSCV}$, $\hat{h}_{BCV}$.

Table 8.1: Table summary of literatures that compares distribution estimation. The third columns are the methods of estimating the bandwidth for kernel methods which are further described in Chapter 2.

In Table 8.1, each literature evaluates the performance differently which includes comparing the estimated MISE ([67], [97] and [151]), comparing the estimated AMISE ([69]) and comparing the shape of the estimated distribution to the true distribution of the simulated data ([91] and [150]).

[151] and [147] ran experiments on distribution learners that compare the learners and evaluates the goodness of the learners by loss functions. [151] further rank the learners from best to worst. However, [151] did not discuss how estimating and evaluation is done specifically with the use resampling method. [147] compared

kernel distribution learner with a discretized classifier. The experiment is to compare existing distribution learners with a new proposed method. Unlike [151], [147] used real world datasets.

Most of the simulation experiments are to compare the proposed methods with existing methods. Furthermore, each simulation experiment uses different methods of evaluation (MISE, log-likelihood and AMISE). Hence it is difficult to make a comparisons using different literatures. For example, [67], [91], [151], [150] included h_{LSCV} in the comparisons but uses different method of evaluation. In addition, the experiments in Table 8.1 also did not specify the resampling method of how the fitting and estimation of the distribution is done.

Comparing distribution estimation learners using ensemble learning methods was further discussed by [71], [153], [48], [80], [79] and [82]. [71] showed that stacked KDE is better when compared to other methods when evaluated using log-likelihood (i.e. negative log-loss) on 4 real world datasets. [153] and [82] compared different bagging PDF estimation learners. [153] found that for classification problem using kernel method, bagging PDF produced the highest log-likelihood (i.e. negative log-loss). [82] compared 3 different PDF estimation method including histogram, frequency polygons and kernel learners with their respective bagging learners. The simulated experiments on 8 simulated datasets shows that each bagged learners produced smaller MISE than the non-bagged learners. [79] showed that as the size of datasets increases, the bagged learners have better performance in terms of MISE. Boosted PDF learners were studied by [48] and [80]. The latter showed that boosted PDF learners are less effective for datasets from a skewed distribution.

8.2 Benchmarking Experiment for Distribution Estimation

In this section, we describe the benchmarking experiment for distribution learners implemented using **mlr3proba**, **mlr3** and **mlr3extralearners**. We start by listing the objectives of the experiment. Then, we outline the design of the experiment by specifying the datasets used, resampling method, learners and evaluating strategy. In this benchmarking experiment, we are not just comparing one type of distribution learner. In addition, we also compare different learners from different family of distribution (kernel, histogram, KNN, penalized and spline). The learners will be grouped with its respective family. For example, KDE plug-in methods are collected together in one group, histogram learners will be grouped together and etc.

8.2.1 Objective of Benchmarking experiment

Here, we outline the objectives of this benchmarking experiment.

i. To compare and rank distribution learners of different family of learners with respect to log-loss.
ii. To compare and rank kernel based distribution learners (including tuned and plug-in parameter) with respect to (1) log-loss; (2) Probabilistic squared loss (PSL)
iii. To compare and rank Gaussian kernel distribution learners with respect to (1) log-loss; (2) Probabilistic squared loss (PSL); (3) Integrated Brier loss (IBL)
iv. To compare and rank the family of distribution learners w.r.t log-loss.

From the above objectives, (ii) is limited to kernel based distribution because there is no functionality in **R** to support non-kernel distribution. (iii) is further limited to Gaussian kernel because the computation of IBL for other kernels are time consuming for large dataset.

8.2.2 Design of Benchmarking Experiment

In this section, we describe the design of the benchmarking experiment for distribution estimation. First, we outline the datasets used for this experiment. Second, we explain the resampling method. Third, we list the algorithms used for the experiment, classifying them into different groups. Finally, we describe the evaluation methods for all algorithms. From running this benchmarking experiment, the out-of-sample empirical generalization losses for each dataset and each learner is obtained. We report the average of out-of-sample empirical generalization losses for each learner over all datasets. The results of this experiment and its analysis are further discussed in Section 8.3.

8.2.2.1 Data for Benchmarking Distribution Estimation

This section is a list of datasets used in this benchmarking experiment on distribution estimation. The datasets used in this benchmarking experiment are from the UCI ([111]) database. The UCI database contain datasets that have been grouped into its respective task, i.e. regression, classification, clustering and others. There is no specific datasets for distribution learners. However, since we are running experiment on univariate distribution estimation, we have an advantage to select any dataset in the database. We restrict our choice to continuous data because we are experimenting learners for continuous setting. We also have the advantage to select multiple variables or attributes within each dataset to make separate datasets. For example, we extract two attributes from Air Quality dataset, temperature and relative humidity as two separate datasets in distribution estimation task. We used in total 54 datasets. These are attributes from 29 separate datasets from UCI database.

In general, we do not perform any preprocessing on the datasets to maintain the originality structure. However, we carry out data-cleaning, to remove `NA` and 0, to ensure continuity. This is done by removing the rows of the dataset that contain `NA` and 0. Therefore, the number of instances (or observations) in the cleaned dataset is less than the original dataset.

We list the datasets below with a brief description on them. We note down the variables extracted as a new dataset, the changes we made on the data and the number of instance (observations) of the dataset used after data cleaning. We use the format (description of the dataset - the attributes or variables used - number of instance).

i. **Airfoil self noise**: Dataset that consists of different size of ‘NACA 0012’ airfoils at various wind tunnel speeds and angles of attack - *scaled sound pressure level, (V6)* - instances 1503.

ii. **Arrhythmia:** Dataset to differentiate the presence and absence of cardiac arrhythimia then categorized into 16 groups- *V248* and *V249* - 452 instances.

iii. **Audit data:** Non-confidential dataset from 2015 to 2016 of firms. The dataset is collected from the Auditor Office of India. The dataset is used to build a predictor for classifying suspicious firm - *Total* and *Audit_risk* - 685 instance; [154] and supported supported by Ministry of Electronics and Information Technology (MEITY), Govt.of India.

iv. **Australian Credit Approval**: Dataset on credit card application in Australia - *A2* and *A3* - 690 instances.

v. **Climate model simulation crashes:** Dataset for predicting climate model simulation crashes. The dataset is used to find the parameter value combinations that cause the failures using Latin hypercube samples of 18 climate model input parameter values - *vconst_corr* and *Prandtl* - 540 instances; [155].

vi. **Cloud**: Dataset on cloud image - *Visible minimum value (V3)* and *IR minimum value (V10)* - 1024 instances.

vii. **Concrete slump test:** Dataset on factors that influenced the concrete - *Comprehensive strength* - 103 instance; [156].

viii. **Concrete Comprehensive Strength Data**: Dataset on concrete strength for civil engineering - *28- day comprehensive strength*- instance 1030; [157].

ix. **Credit approval:** Data about credit card application - *V2* and *V3 attributes* - 671 instances.

x. **Dow Jones Index:** Dataset of 6 week data of Dow Jones Industrial Index - *percentage return dividend* - 750 instances; [158].

xi. **Energy efficiency**: Dataset for assessing the heating and cooling load requirements. The data is used to build a function - *cooling load (Y1)* and *heating load (Y2)* - 768 instances; [159].

xii. **Forest fire**: Dataset on meteorological and other data. The dataset is used to predict the burned area of forest fires in Portugal - *FFMC*, *DMC* and *MC* - 517 instances; [160]

xiii. **Heart failure clinical record**: Dataset consists of patients' medical record who have heart failure with 13 features - *serum creatinine* -299 instances; [161]

xiv. **Glass Identification:** Dataset of 6 types of glasses - *Magnesiu, V4*, *Aluminum, V5* and *Silicon, V6* - 172 instaces.

xv. **Istanbul Stock Exchange**: Dataset of returns of Istanbul Stock Exchange with 7 international index - *ISE* and *EM* - 536 instances; [162].

xvi. **HCV data:** Dataset on laboratory values of blood donors and Hepatitic C patients and demographic value - *CHE* - 615 instances.

xvii. **Ionosphere:** Dataset on radar from the ionosphere - *V4* - 615 instances.

xviii. **Leaf:** A dataset of collection of shapes and texture features extracted from digital images if leaf specimen from 40 different plants - *Elongation, (V5)*, *Average Intensuty, (V11)* and *Entropy, (V16)* - 340 instances; [163].

xix. **Meta-data:** Dataset for classification method - *Continuous error, (V22)*.

xx. **Mice protein expression:** Dataset consist of 77 proteins - *Ubiquitin_N level* and *pCAMKII_N level* - 1080 and 1077, respectively; [164].

xxi. **Parkinson's dataset:** Dataset from Oxford's Parkinson's Disease detection - *RPDE* and *PPE* - 195 instances; [165].

xxii. **QSAR aquatic toxicity**: Dataset used to predict quantitative acute toxicity - *Molecular properties (MLOGP)*, *2D autocorrelation (GATS1p)* and *aquatic toxicity (LC50)*, (total 3) - 546 instances ; [166].

xxiii. **QSAR Bioconcentration classes**: Dataset for QSAR modeling the manually-curated Bioconcentration factor (BCF, fish) and mechanistic classes - *MLOGP* and *LogBF* - 774 instances; [167] [168]. instances 908 [169]

xxiv. **Real estate valuation data set**: Dateset consists of historical real estate evaluation from Taiwan - *X3, the distance to the nearest MRT station* and *Y1, house price of unit area* - 414 instances; [170].

xxv. **Seeds:** Dataset on the measurements of geometrical properties of kernels for three different varieties of wheat - *width of kernel, (V4)*, *asymmetry coefficient, (V5)* and *length of kernel groove, (V6)* - 210 instances; [171] with support from Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

xxvi. **Synthetic control chart time series:** Dataset on synthetic generated control charts - *Upward shift, (V5)* - 600 instance; (Eamonn Keogh).

xxvii. **Vertebral column:** Dataset to classify orthopaedic patients - *V1*, *V2*, *V5* and *V6* - 310 instances.

xxviii. **Wine:** Dataset of chemical analysis to discover wine origin - *Malic acid, (V2)* and *Nonfalvanoid acid, (V8)* - 178 instances.

xxix. **Wisconsin breast cancer (diagnostic):** Dataset of diagnostic Wisconsin breast cancer - *V8*, *V9* and *V31* - 556 instances,

xxx. **Yacth hydrodynamics:** Dataset to predict the performance of hydrodynamic of sailing yachts using dimensions and velocity - *Residuary resistance per unit weight of displacement (V7)* - 308 instances.

8.2.2.2 Resampling Method

For this benchmarking experiment, we use a 3-fold cross-validation resampling method. For each dataset, we split into 3 parts (see Chapter 6 on K-fold cross-validation). The first fold is used as the test dataset while the remaining two folds are for fitting (training). This process is repeated until all folds are used as the test datasets.

8.2.2.3 Evaluation

In this experiment, the learners are evalauted using the expected generalization loss. First, the out-of-sample empirical generalization loss is computed to evaluate the learners on each fold using the 3-fold cross-validation resampling method using Algorithm 17. Then, the out-of-sample empirical generalization loss is averaged over the 3 folds to obtain the *mean empirical loss*. For all learners, we compute the *mean empirical log-loss*. For all kernel based method, we compute the *mean empirical PSL* and lastly for Gaussian kernel based learners, we compute the *mean empirical IBL.*

8.2.2.4 Learners for Benchmarking Distribution Estimation

In this section, we provide a list of learners used in this benchmarking experiment. The distribution estimation learners that will be compared are available in **R** and collected in **mlr3proba** and **mlr3extralearners**. Some of these algorithms require tuning and some use the default parameter. Distribution estimation task is well-researched area and there are many methods being introduced. However, in choosing the learners to compare, we select the learners that have already been implemented in **R** and tuned learners. To achieve our first objective, we use not only kernel based learners but also include different families of learners. The learners are grouped into two big families: (1) kernel based; (2) non-kernel based. For (1), we further split into tuned algorithms and plug-in algorithms. For (2), we split into smaller groups based on the family of learners.

Kernel distribution learners

Here, we consider the kernel based distribution learners.

Out-of-sample Tuned Learners: The out-of-sample method is based on the discussion from Chapter 6. Here, we describe the specific tuning method for this benchmarking experiment including the resampling, the vector of bandwidth used and the loss function for optimizing.

Firstly, we used a grid search method for this experiment. We set the values of the bandwidth to be from 0.1 to 10.

Secondly, in selecting a tuned model, this is done by minimising the empirical loss w.r.t the bandwidth. We used three loss function: (1)log-loss; (2) PSL; (3) IBL.

Thirdly, we set up the resampling strategy for this tuning method. We use a 3-fold cross-validation to resample the training set. For each pair of inner training and inner test sets, we use Algorithm 18 to obtain the tuned learner (reflecting the optimal parameter for the learner) and the minimum out-of-sample empirical generalization loss. For each fold, the bandwidth with the smallest out-of-sample empirical generalization loss is selected and used for prediction (refer to Figure 6.3).

In total, we will have 12 out-of-sample tuned learners which are listed below.

i. **dens.kde.gaus.ll**: A kernel PDF learner that uses Gaussian kernel in which the bandwidth is tuned by grid search using log-loss.
ii. **dens.kde.gaus.psl**: A kernel PDF learner that uses Gaussian kernel in which the bandwidth is tuned by grid search using PSL.
iii. **dens.kde.gaus.ibl**: A kernel PDF learner that uses Gaussian kernel in which the bandwidth is tuned by grid search using IBL.
iv. **dens.kde.epan.ll**: A kernel PDF learner that uses Epanechnikov kernel in

which the bandwidth is tuned by grid search using log-loss.

v. **dens.kde.epan.psl**: A kernel PDF learner that uses Epanechnikov kernel in which the bandwidth is tuned by grid search using PSL.

vi. **dens.kde.unif.ll**: A kernel PDF learner that uses Uniform kernel in which the bandwidth is tuned by grid search using log-loss.

vii. **dens.kde.unif.psl**: A kernel PDF learner that uses Uniform kernel in which the bandwidth is tuned by grid search using PSL.

viii. **dens.kde.quart.ll**: A kernel PDF learner that uses Quartic kernel in which the bandwidth is tuned by grid search using log-loss.

ix. **dens.kde.quart.psl**: A kernel PDF learner that uses Quartic kernel in which the bandwidth is by tuned grid search using PSL.

Plug-in Kernel Based Algorithms:

i. **kdeKD.gaus**: A kernel PDF learner that uses Gaussian kernel from package **kerdiest** ([172]) with the plug-in method of bandwidth, h with $h = h_{PB}$ where h_{PB} as in Eqn (4.2.43); [97].

ii. **kdeKD.epan**: A kernel PDF learner that uses Epanechnikov kernel from package **kerdiest** ([172]) with the plug-in method of bandwidth, h with $h = h_{PB}$ where h_{PB} as in Eqn (4.2.43); [97].

iii. **kdeKD.quar**: A kernel PDF learner that uses Quartic kernel from package **kerdiest** ([172]) with the plug-in method of bandwidth, h with $h = h_{PB}$ where h_{PB} as in Eqn (4.2.43); [97].

iv. **kdeKS.gaus**:A kernel PDF learner that uses Gaussian kernel from package **ks** ([173]) with the plug-in method by [68] in Eqn (4.2.22).

v. **kdeSM.gaus**: A kernel PDF learner that uses Gaussian kernel from package **sm** ([174]) with the plug-in method as in Eqn (4.2.13); [132], [5].

vi. **kdeNP.gaus:** A kernel PDF learner that uses Gaussian kernel from package **np** ([175]) with the default cross-validation using maximum likelihood.

vii. **kdeNP.epan:** A kernel PDF learner that uses Epanechnikov kernel from package **np** ([175]) with the with the default cross-validation using maximum likelihood estimation.

viii. **kdeNP.unif:** A kernel PDF learner that uses Uniform kernel from package **np** ([175]) with with the default cross-validation using maximum likelihood.

ix. **kde.norm:** A kernel PDF learner that uses Gaussian kernel using h_{ROT}, Eqn (4.2.14), [5]..

x. **kde.epan:** A kernel PDF learner that uses Epanechnikov kernel using h_{ROT}, Eqn (4.2.14), [5].

xi. **kde.unif:** A kernel PDF learner that uses Uniform kernel using h_{ROT}, Eqn (4.2.14), [5].

xii. **kde.quart:** A kernel PDF learner that uses Quartic kernel using h_{ROT}, Eqn (4.2.14), [5].

Non-kernel Based Algorithms

Here, we consider non-kernel based distribution learner that we have introduced in the previous chapter.

Histogram:

i. **dens.hist.sturges:** Estimate the PDF using `hist` in **graphic** package using the default method Sturges' rule, with the number of bins, $B = \log(N) + 1$, where N is the number of training data points.

ii. **dens.hist.scott:** Estimate the PDF using `hist` in **graphic** package using Scott's method, the bin width $w = \frac{3.49s}{\sqrt[3]{N}}$, with N is the number of training data points.

iii. **dens.hist.bin:** Estimate the PDF using the `hist` in **graphic** package but tuning the **number** of bins (input is a number) via log-loss. The tuning follows the same structure as out-of-sample tuned kernel learners.

KNN Density Methods:

i **dens.knn.sil**: Estimate the PDF using KNN from package **TDA** ([176]) using the function `knnDE` and Silverman's rule of thumb as in [5] with the number of k nearest neighbours, $k = N^{1/2}$, where N is the number of instances (observations).

ii **dens.knn.kung**: Estimate the PDF using KNN from package **TDA** ([176]) using the function `knnDE` and parameter proposed by [10] with the number of k nearest neighbours, $k = N^{1/d}$ where N is the number of observations and d is the dimension.

Penalized Method:

i. **dens.pen.gaus**: Estimate PDF via penalized mixture approach ([177], [178]).

ii. **dens.logspline**: Estimate the PDF using logspline approach proposed by [18].

8.3 Results and Discussion

In this section, we discuss the result of the benchmarking experiment. In total, this experiment consist of 28×54 results of log-loss, 24×54 results of PSL and 8×54 results of IBL to evaluate on. Recall from Section (8.2.2.1), some of the datasets are attributes from the one dataset. Therefore, we need to consider the *independence* of the result when analysing the result of this experiment. This is done by averaging the result which is discussed later on hypothesis testing using Friedman test. We first explain what is the output of the benchmarking experiment, what methods we use to analyse the benchmarking experiment and how we use the methods.

Output of the Benchmarking Experiment

The benchmarking experiment produced tables of results that reports the following:

i. Using log-loss as the evaluation method, it has 54 rows (number of datasets) and 28 columns (number of learners) of averaged out-of-sample empirical log-loss.
ii. Using PSL as the evaluation method, it has 54 rows (number of datasets) and 21 columns (number of learners) of averaged out-of-sample empirical PSL.
iii. Using IBL as the evaluation method, it has 54 rows (number of datasets) and 8 columns (number of learners) of averaged out-of-sample empirical IBL.

Methods for Analysing

Here, we list down the methods used for analysis, describing how to use the methods for analysing our results and the purpose of the analysis.

i. **Average loss**: For each dataset and each learner, we obtained results of 3 out-of-sample empirical loss resulting from 3-fold cross-validation. The average of the out-of-sample empirical loss is computed which depends on the learner and dataset. We call this *mean empirical loss*. Then, for each learner, the average of *mean empirical loss* over all the datasets is computed. We call this as *average loss*. The best learner is chosen with the minimum average loss.
ii. **Ranking the learners**: For each dataset, we rank the learners. The learner with the minimum *mean empirical loss* has the lowest average rank value while the learner with the learner with the maximum *mean empirical loss* has the highest average rank. Then, the average rank of each learner over all 54 datasets is computed and we call this as *average rank*. Using this ranking system, the best learner has the lowest average rank while the worst performing

learner has the a highest average rank.

iii. **Hypothesis test:** We conduct a hypothesis test to compare whether there is significant difference between the learners as proposed by [179]. We need to consider independence for this test because some of the datasets are related (i.e. some of the dataset are attributes from the same original datasets). We use the following test:.

   a. **Friedman test ([180])**: Friedman test is used when independence is assume for each cell in the data. In this benchmarking experiment, some datasets are a related (because some of the datasets are attributes from a similar dataset). For example, the variables of dataset **Seeds** are taken as separate datasets. To overcome this non-independence, the mean empirical loss for the datasets (which are attributes to Seeds) '*width of kernel*', '*asymmetry coefficient*' and '*length of kernel groove*' are averaged for each distribution learner. This is done for all related datasets. This reduce our datasets from 54 to 30 (the new table of result is now 28×30 for using log-loss to evaluate and 24×30 and 8 for PSL and IBL, respectively). However, Friedman test is considered an advantage in our experiment because no assumption is made about the distribution of our results (i.e. Friedman test is non-parametric). The hypotheses for Friedman test are:
   Null hypothesis: There is no significant difference between the learners.
   Alternative hypothesis: There is a significant difference between the learner.
   For Friedman-test, if the null hypothesis is rejected, we proceed to the post-hoc test that test for significant difference between pair learners.

   b. **Nemenyi test ([181]):** This is a post-hoc test after the Friedman test results in a significant difference. The Nemenyi test compares all the learners with each other. A critical difference value is computed (see [179]). If the difference of average rank between two learners is greater than CD then we can conclude that the two learners are significantly different.

   c. **Critical difference (CD) diagram ([179]):** To visualise the Nemenyi test, [179] proposed to plot a CD diagram. In the plot, the learners are ranked increasingly. The learner with the smallest average loss has the smallest rank value whereas the learner with the largest average loss has the highest rank value. Learners that are not significantly different from each other are connected by a horizontal line (i.e. the length of the horizontal line is less than the CD value).

8.3.1 Discussion: Evaluation Using Log-loss

This section is a discussion of analysis of the result for using the log-loss as the evaluation method. This is align to our first objective (see Section 8.2.1).

Firstly, by analysing the *average log-loss* from Table 8.2, kdeKD.Gaus is the best learner out of the 28 learners being compared with average loss of 0.9893. Though we do not consider any learner to be a baseline, it is interesting to note that that kdeKD.gaus has a lower *average log-loss* compared to kde.gaus which uses the Silverman's rule of thumb and Gaussian kernel. Out-of-sample tuned learners did not performed well when evaluated using the log-loss with kde_norm_psl.tuned has highest average loss of 3.894. Overall, kernel methods using bandwidth as in Eqn (4.2.43) have a lower *average log-loss* compared to all other methods.

In terms of average ranking, dens.knn.sil (i.e. KNN learner using Silverman's rule) is rank first with average rank of 3.59 but dens.knn.kung (i.e. KNN learner using [10]'s rule) is rank last with average rank of 25.33. The average rank does not align with the result of average log-loss. This shows that the performance of the distribution learners depends on the dataset. dens.knn.sil performs worst than other learners on the ionosphere dataset with mean empirical log-loss of 1.7932. However, that is the only dataset that show's it's worst performance. For other datasets, dens.knn.sil's average rank is 5. On the other hand, the rank of kdeKD.gaus which is the best learner with respect to average log-loss, fluctuates. Out 54 datasets, it ranks first for 29 datasets. However, it ranks 26 for 10 datasets. Leading its average rank to 10.04.

Learners	Log-loss	Rank	Learners	Log-loss	Rank
dens.knn.sil	1.793 (5)	3.59	kdeNP.epan	1.997 (8)	10.50
dens.knn.kung	2.877 (26)	25.33	kdeKS.gaus	1.709 (4)	8.67
dens.hist.sturges	2.189 (18)	16.72	kdeSM.gaus	2.075 (11)	14.17
dens.hist.scott	2.154 (17)	16.80	dens.kde.gaus.ll	2.227 (21)	15.25
dens.hist.bin	2.208 (19)	18.67	dens.kde.unif.ll	2.214 (20)	17.56
kde.gaus	2.061 (10)	12.56	dens.kde.epan.ll	2.111 (15)	14.26
kde.unif	2.078 (13)	14.20	dens.kde.quart.ll	2.134 (16)	13.99
kde.epan	2.089 (14)	15.00	dens.kde.gaus.psl	3.894 (28)	17.64
kde.quart	2.078 (12)	14.44	dens.kde.unif.psl	2.885 (27)	19.58
kdeKD.gaus	0.989 (1)	10.04	dens.kde.epan.psl	2.602 (24)	17.02
kdeKD.epan	1.021 (2)	11.02	dens.kde.quart.psl	2.657 (25)	16.40
kdeKD.quart	1.024 (3)	12.02	dens.kde.gaus.ibl	2.347 (22)	16.69
kdeNP.gaus	1.996 (7)	9.80	dens.pen.gaus	2.496(23)	23.43
kdeNP.unif	2.033 (9)	12.31	dens.logspline	1.835 (6)	8.35

Table 8.2: Table of average log-loss from benchmarking experiment. The column of 'log-loss' is the average log-loss with the rank based on the average log-loss in the bracket. 'Rank' column refers to the average rank as obtained by ranking the learners above.

The average rank for the learners compared using average log-loss is shown in Figure 8.1 with the dens.knn.sil is ranked first while dens.knn.kung is ranked last.

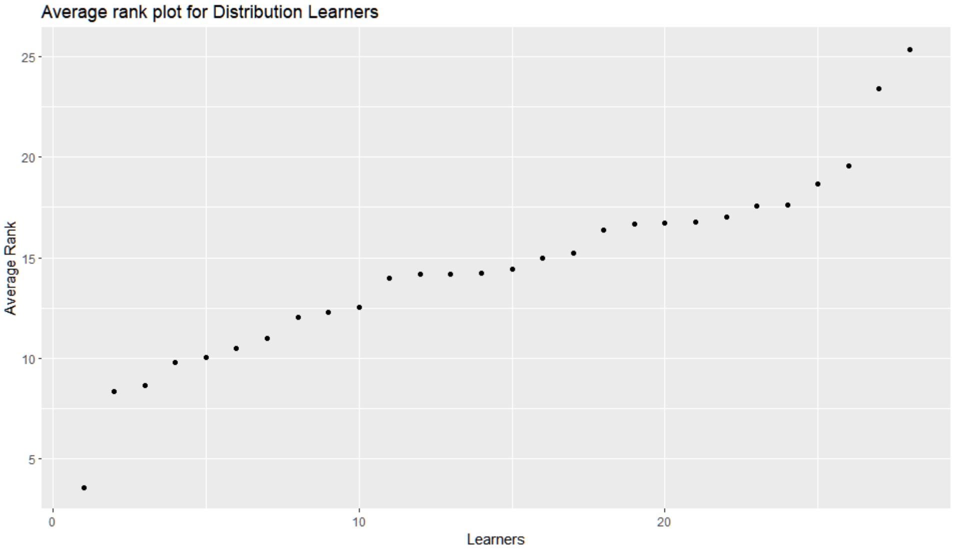


Figure 8.1: Plot of average rank of learners evaluated using log-loss. The x-axis is the learners in an increasing order: dens.knn.sil, dens.logspline, kdeKS.gaus, kdeNP.gaus, kdeKD.gaus, kdeNP.epan, kdeKD.epan, kdeKD.quart, kdeNP.unif, kde.norm, dens.kde.quart.ll, kdeSM.gaus, kde.unif, dens.kde.epan.ll, kde.quart, kde.epan, dens.kde.gaus.ll, dens.kde.quart.psl, dens.kde.gaus.ibl, dens.hist.sturges, dens.hist.scott, dens.kde.epan.psl, dens.kde.unif.ll, dens.kde.gaus.psl, dens.hist.bin, dens.kde.unif.psl, dens.pen.gaus, dens.knn.kung. The y-axis is the average rank of each learners.

Friedman test results in rejecting the null hypothesis with p-value 1.63×10^{-42} which is very low. Concluding there is a significant difference between the learners. Post-hoc test for Friedman test using Nemenyi test suggested there are significant difference between the learners. The critical difference (CD) using average log-loss is 8.024.

From Figure 8.2, there are 8 groups of distribution learners. Each group is connected by a horizontal line. Within each group, the learners are not significant different of each other (the differences of the ranks between two learners are less than the critical difference). Two learners where the difference of average rank exceed the CD value (8.024) are significantly different. For example, dens.KNN.sil is significantly different than kdeNP.unif and dens.logspline is significantly different than dens.kde.quart.psl.

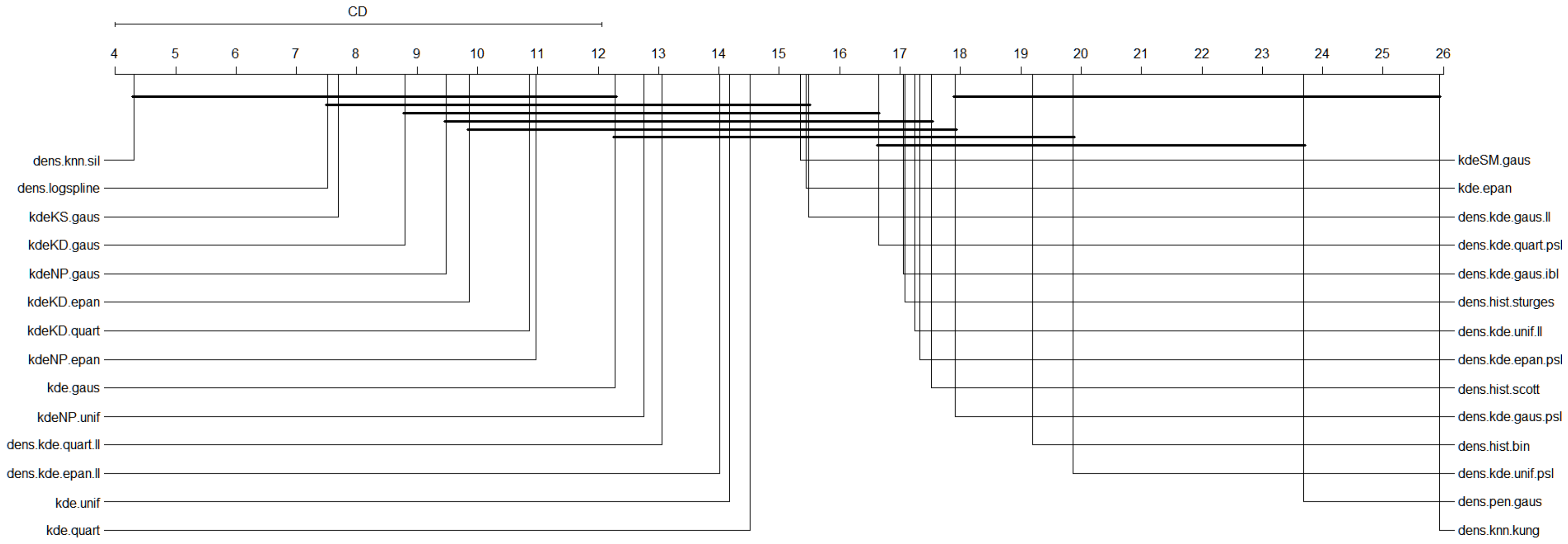


Figure 8.2: Critical difference (CD) diagram from the result of Nemenyi post-hoc test for using log-loss as the evaluation method. The learners are ranked increasingly. The learners with the lowest average rank are better than those with a the higher average rank. The critical difference (CD) is 8.024. The horizontal lines show groups of learners that are not significantly different.

Comparisons Between Family of Distribution Learners

In this section, we analyse the results of the experiment to obtain the fourth objective, which is to compare and rank the family of distribution learners using log-loss. We grouped the learners based on Section 8.2.2.4. In total, we have 5 groups of family of learners: (1) tuned kernel methods; (2) plug-in kernel method; (3) KNN distribution learners; (4) histogram learners; (5) penalized distribution learners. To analyse this, we only compare the average loss and the average rank.

i. The average log-loss is computed by averaging the mean empirical log-loss over the learners of the same family and all datasets. For example, the average log-loss of penalized learner family in Table 8.3 is computed by averaging the mean empirical log-loss over all 54 datasets and over two learners, dens.pen.gaus and dens.logspline.
ii. For the average rank, we assign the rank of each learner based on the mean empirical log-loss. Then, average them over 54 datasets and its family of learners. For example, the average rank for penalized family in Table 8.3 is obtained by averaging the rank based on mean empirical log-loss over 54 datasets and two learners, dens.pen.gaus and dens.logspline.

Due to the above, running a hypothesis test is not suitable as it violates the independence requirement of the Friedman test.

In terms of average log-loss, the plug-in kernel methods have a lower *average log-loss* compared to all other methods with the average loss of 1.7625. This is not surprising as based in Table 8.2, the kdeKD.gaus, kdeKD.epan and kdeKD.quart have lower mean empirical log-loss compared to others. Whereas, the family of tuned kernel methods are have the highest average loss, 2.5633. From Table 8.2, the dens.kde.gaus.ll has a high average log-loss of 2.227. This might be the caused for the high average log-loss of 2.5633 for tuned learners. Family of KNN, histogram and penalized methods have the average loss of 2.3352, 2.1839 and 2.1657, respectively.

The average rank of the family in ascending order is plug-in kernel (2.11), Hist (2.67), penalized (3.09), Tuned kernels (3.20) and KNN(3.93). In terms of average rank, the family of plug-in kernel methods have a lower *average log-loss* of 2.11 compared to other methods. But the KNN methods are rank the highest with 3.93 with tuned kernels methods still in the upper half with 3.20.

Learner Family	**Average loss**	**Rank**
Tuned kernel learners	2.563 (5)	3.20
Plug-in kernel learners	1.763 (1)	2.11
KNN	2.335 (4)	3.93
Histogram	2.184 (3)	2.67
Penalized	2.166 (2)	3.09

Table 8.3: Table of average log-loss and average rank for family of learners w.r.t log-loss. The column 'average' log-loss is computed by averaging the mean empirical log-loss over the learners of the same family and all datasets. The numbers in the bracket show the rank based on the average loss. The column 'Rank' is computed by averaging the rank based on mean empirical log-loss over all datasets and respective family.

8.3.2 Discussion: Evaluation Using PSL

In this section, we will discuss the results for evaluating distribution learners using PSL. Due to computing limitation, PSL is only used to evaluate kernel learners. Therefore, we conduct the benchmarking experiment to compare the 21 kernel distribution learners. The results are shown in Table 8.4.

Learners	PSL	Rank	Learners	PSL	Rank
kde.gaus	-1.827 (5)	9.53	kdeSM.gaus	-1.791 (7)	12.88
kde.unif	-1.837 (2)	9.44	dens.kde.gaus.ll	-0.760 (18)	13.24
kde.epan	-1.833 (3)	9.80	dens.kde.unif.ll	-0.812 (15)	13.70
kde.quart	-1.833 (4)	9.72	dens.kde.epan.ll	-1.058 (12)	12.31
kdeKD.gaus	0.841 (19)	7.59	dens.kde.quart.ll	-1.029 (13)	11.48
kdeKD.epan	0.942 (20)	8.69	dens.kde.gaus.psl	-0.774 (16)	10.49
kdeKD.quart	0.965 (21)	9.64	dens.kde.unif.psl	-0.854 (14)	11.69
kdeNP.gaus	-1.827 (6)	9.81	dens.kde.epan.psl	-1.099 (10)	10.17
kdeNP.unif	-1.682 (8)	14.06	dens.kde.quart.psl	-1.069 (11)	9.89
kdeNP.epan	-1.645 (9)	15.94	dens.kde.gaus.ibl	-0.773 (17)	12.62
kdeKS.gaus	-1.846 (1)	8.31			

Table 8.4: Table of average PSL from benchmarking experiment. The column 'PSL' is the average PSL with the rank in the bracket. The column 'Rank' refers to the average rank as obtained by ranking the learners above.

In terms of average PSL, the best learners is chosen with the minimum *average PSL*. Overall, kdeKS.gaus is the best learner with the average PSL of -1.8458 while kdeKD.quart is the least performed learner with the average PSL of 0.965. All of Silverman's rule of thumb method (kde.gaus, kde.unif, kde.epan and kde.quart) perform better than other learners. The tuned learners obtained by out-of-sample tuning using log-loss, PSL and IBL only perform averagely.

The average ranking from evaluating using PSL is plotted in Figure 8.3. In terms of average rank, kdeKD.gaus is ranked first with average rank of 7.59, and kdeKS.gaus maintain its position as the top two learners with the average rank of 8.31. Silverman's rule of thumb learners are ranked higher than kdeKD learners. kde.gaus.ll does not performed in terms of rank as its average rank is 13.24.

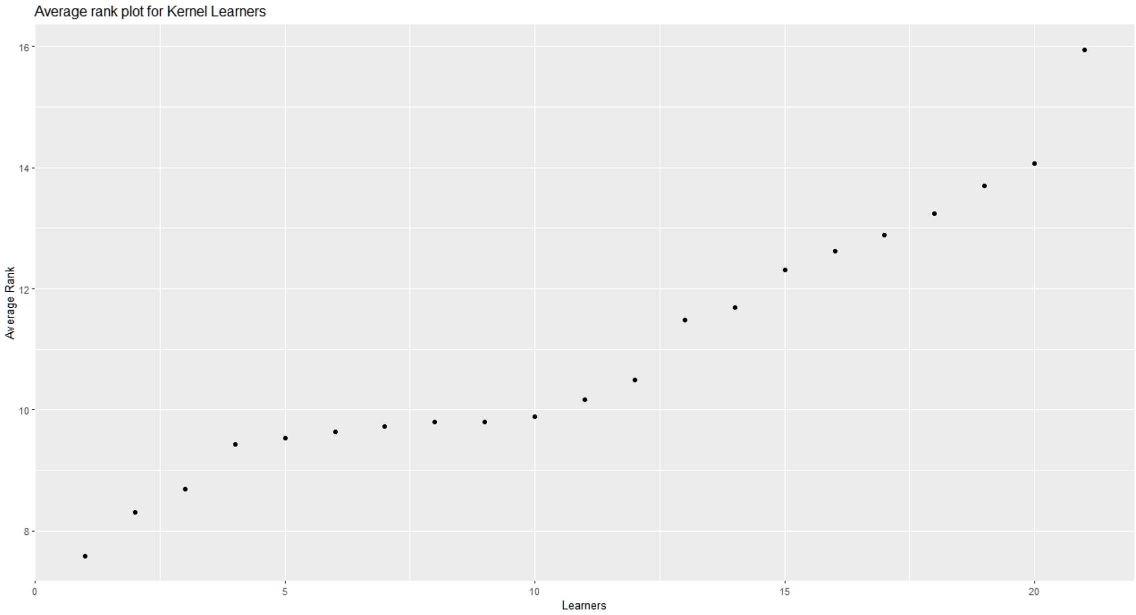


Figure 8.3: Plot of average rank of learners evaluated using PSL. The x-axis is the learners in an increasing order: kdeKD.gaus, kdeKS.gaus, kdeKD.epan, kde.unif, kde.gaus, kdeKD.quart, kde.quart, kde.epan, kdeNP.gaus, dens.kde.quart.psl, dens.kde.epan.psl, dens.kde.gaus.psl, dens.kde.quart.ll, dens.kde.unif.psl, dens.kde.epan.ll, dens.kde.gaus.ibl, kdeSM.gaus, dens.kde.gaus.ll, dens.kde.unif.ll, kdeNP.unif, kdeNP.epan. The y-axis is the average rank obtained by ranking the learners of each learners for each dataset and averaged the rank over all the datasets for each learners.

Friedman test shows there is a strong significant difference between the learners with p-value of 8.27×10^{-11} ($\chi_{20}(89.727)$). Post-hoc test for Friedman test using Nemenyi test suggest there are significant difference between learners when comparing them pairwise. The critical differences (CD) is 5.816 and the CD diagram for PSL is shown in Figure 8.4.

From Figure 8.4, kdeKD.gaus is significantly different from kdeNP.epan. kde.KD learners are ranked better than other learners, taking the top 4 rank. Furthermore, kdeKS.gaus does not show much difference than the result from *average rank* as it ranks 2nd. From Figure 8.4, kdeKS.gaus is also better than kde.gaus in terms of ranking. Learners using Silverman's rule of thumb are ranked averagely while tuned learners did not performed better than most plug-in learners.

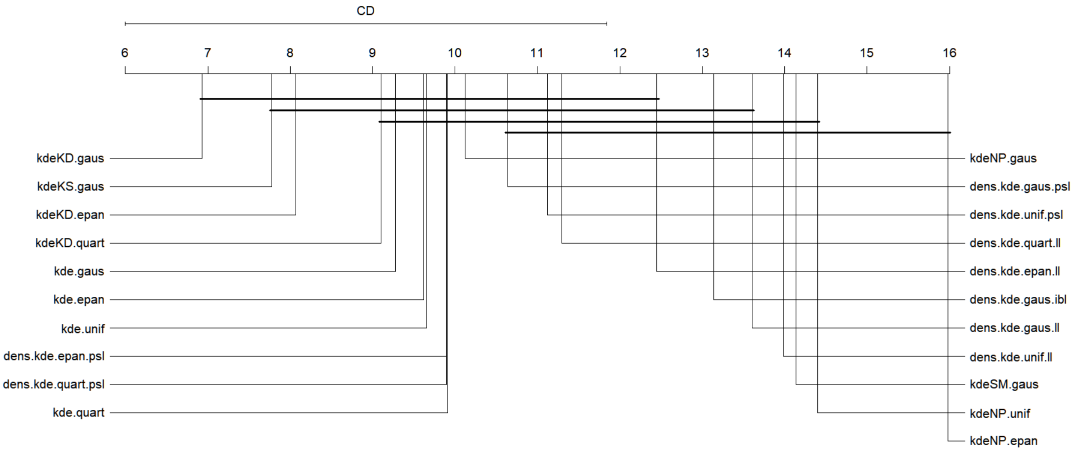


Figure 8.4: Critical difference (CD) diagram from the result of Nemenyi post-hoc test for using PSL as the evaluation method. The learners are ranked increasingly. The learners with the lowest average rank are better than those with a higher average rank. The critical difference (CD) is 5.816. The horizontal lines show groups of learners that are not significantly different.

Comparison Between Tuned kernel Learners

This section is a discussion on comparing of the performance of the tuned learners using PSL.

In terms of average PSL, dens.kde.epan.psl is the best performing learner with the average loss of -1.0985 followed by dens.kde.quart.psl with average PSL of -1.0692. dens.kde.gaus.ll has the highest average PSL -0.760 and dens.kde.gaus.ibl has the second highest average PSL of -0.773.

By observing the average rank between the learners as in Table 8.5, dens.kde.quart.psl rank first with average rank of 3.58 while dens.kde.epan.psl. has average rank of 4.111 and dens.kde.gaus.ll has average rank of 6.08. The tuned methods via PSL rank higher than the tuned methods via log-loss and IBL when comparing the average rank via PSL.

Learner	Rank	Learner	Rank	Learner	Rank
dens.kde.gaus.ll	6.08	dens.kde.quart.ll	4.62	dens.kde.unif.psl	5.04
dens.kde.unif.ll	6.20	dens.kde.gaus.ibl	5.56	dens.kde.epan.psl	4.11
dens.kde.epan.ll	5.30	dens.kde.gaus.psl	4.42	dens.kde.quart.psl	3.68

Table 8.5: Table of average rank for tuned distribution kernel methods evaluated using PSL. The column 'Rank' refers to the average rank obtained by ranking the learners with respect to mean empirical PSL and average over all 54 datasets.

Friedman test indicates a strong significant difference with p-value of 0.000239 ($\chi^2_8 = 29.7$). To investigate further which pairs of learners that resulted in the difference, we conduct Nemenyi test. The critical difference is 2.231 and the learners that are significantly different are shown in Figure 8.5. The result from Nemenyi test indicates that dens.kde.quart.psl and dens.kde.gaus.ll are significantly different and indicates that dens.kde.quart.psl is better than other tuned learners when evaluated using PSL. It is also interesting to note that dens.kde.quart.ll is rank higher than dens.kde.gaus.ibl.

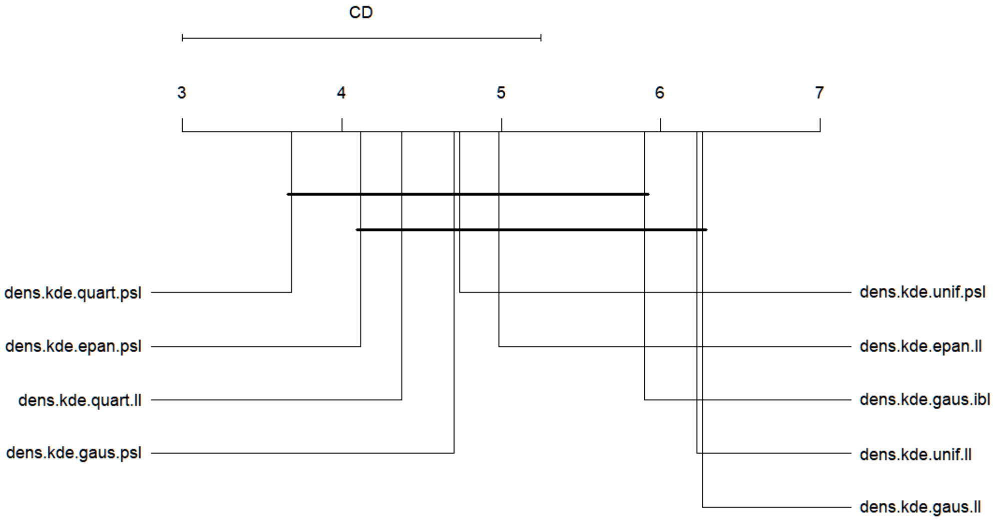


Figure 8.5: Critical difference (CD) diagram from the result of Nemenyi post-hoc test for using PSL as the evaluation method . The learners are ranked increasingly. The learners with the lowest average rank is better than those with a higher rank. The critical distance (CD) is 2.231. The horizontal lines show groups of learners that are not significantly different.

Comparisons between Plug-in Kernel Methods by PSL

For plug-in kernel learners evaluated using PSL, the best learner by analysing the average PSL is the kdeKS.gaus learner with average loss -1.8458 while the least performing learner is kdeKD.quart withe average PSL 0.9646. All distribution learners using Silverman's rule of thumb method have smaller average PSL compare to kdeKD learners followed by kdeNP learners.

The average rank of the plug-in learners using PSL is tabulated in Table 8.6. The average rank is also plotted in Figure 8.6. By analysing the average rank of PSL for plug-in learners, kdeKD.gaus is rank the lowest with average rank of 4.31 whereas kdeNP.epan is rank the highest with average rank 9.78. All of Silverman's rule of thumb are rank averagely with 5.89, 5.95, 6.01 and 6.23 for kde.unif, kde.gaus, kde.epan, and kde.quart, respectively.

Learner	**Rank**	**Learner**	**Rank**	**Learner**	**Rank**
kde.gaus	5.95	kdeKD.gaus	4.31	kdeNP.unif	8.69
kde.unif	5.89	kdeKD.epan	5.34	kdeNP.epan	9.78
kde.epan	6.01	kdeKD.quart	6.29	kdeKS.gaus	5.25
kde.quart	6.23	kdeNP.gaus	6.28	kdeSM.gaus	7.97

Table 8.6: Table of average rank for plug-in distribution kernel methods evaluated using PSL. The column 'Rank' refers to the average rank obtained by ranking the learners with respect to mean empirical PSL and average over all 54 datasets.

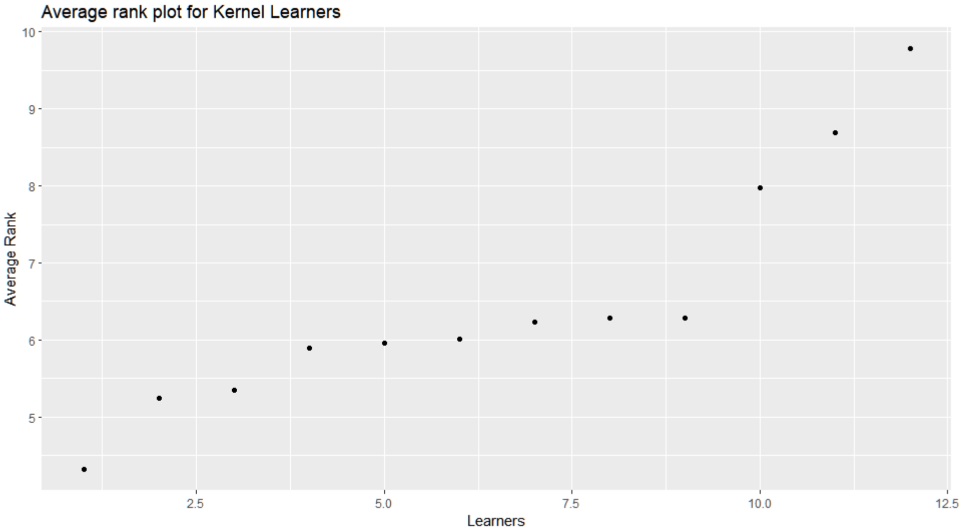


Figure 8.6: Plot of average rank of plug-in learners evaluated using PSL. The x-axis is the learners in an increasing order is: kdeKD.gaus, kdeKS.gaus, kdeKD.epan, kde.unif, kde.gaus, kde.epan, kde.quart, kdeNP.gaus, kdeKD.quart, kdeSM.gaus, kdeNP.unif, kdeNP.epan. The y-axis is the average rank obtained by ranking the mean PSL of each learners for each dataset and averaged the rank over all the datasets for each learners.

Using Friedman test indicates there is a significant difference between the learners where the p-values reported is $2.787{\times}10^{-10}$ ($\chi^2_{11} = 68.112$). The post-hoc test using Nemenyi test also support that there is a difference where the difference between pairs of learners as shown in the CD diagram in Figure 8.7.

From Figure 8.7, kdeKD.gaus learner is significantly different compared to kdeNP.epan. kdeKD.gaus is also ranked better than other kernels whereas kdeKD.epan is ranked third and kdeKD.quart is rank 6. All of Silverman's rule of thumb method ranked between 4 - 7.

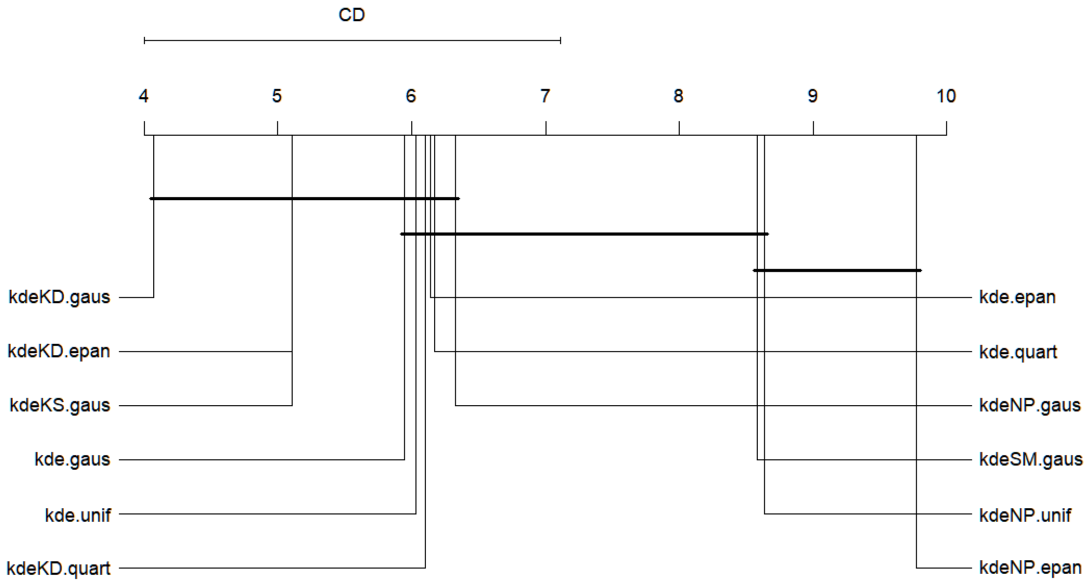


Figure 8.7: Critical difference (CD) diagram from the result of Nemenyi post-hoc test for using PSL as the evaluation method. The learners are ranked increasingly. The learners with the lowest rank is better than higher rank. The critical distance (CD) is 3.094. The horizontal lines show two learners that are significantly different.

8.3.3 Discussion for IBL

In this section, we discuss the results for evaluating distribution learners specifically Gaussian kernel estimators using the IBL. This limitation is due to the high computation time for other kernel learners. Therefore, this benchmarking experiment is able to compare 9 Gaussian kernel learners. The result of the benchmarking experiment evaluated using IBL is shown in Table 8.7.

From Table 8.7, dens.kde.gaus.ibl is the best learner out of the Gaussian kernel learners with the minimum average IBL of 19.704 whereas dens.kde.gaus.ll is rank the last with maximum average IBL of 22.442. Both plug-in methods, kdeKD.gaus and kde.gaus have average IBL of 20.559 and 20.562, respectively. The average rank of the learners evaluated using IBL is shown in Figure 8.8. From analysing the average rank, kdeKS.gaus is the best performed learner whereas dens.kde.gaus.ll is the least performed learner. The result from the average IBL and the average rank of IBL are not consistent.

Learners	IBL	Rank	Learners	IBL	Rank
kde.gaus	20.562 (5)	4.01	kdeSM.gaus	21.373 (7)	4.34
kdeKD.gaus	20.559 (4)	4.27	dens.kde.gaus.ll	22.442 (8)	5.83
kdeNP.gaus	21.083 (6)	3.77	dens.kde.gaus.psl	19.737 (2)	5.66
kdeKS.gaus	20.442 (3)	3.65	dens.kde.gaus.ibl	19.704 (1)	4.46

Table 8.7: Table of average IBL from benchmarking experiment. The column 'IBL' refers to the average IBL. The number inside the bracket is the rank based on average IBL. The column 'Rank' refers to the average rank as obtained by ranking the learners above.

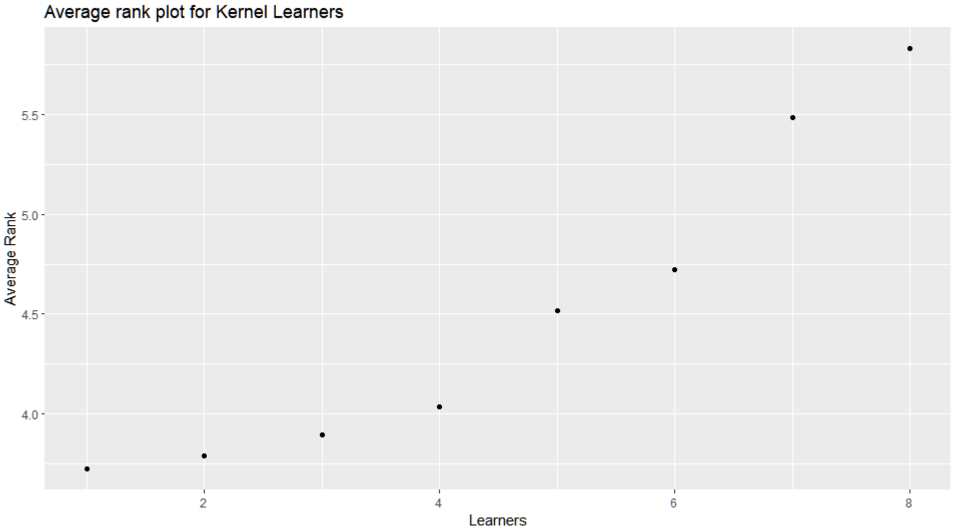


Figure 8.8: Plot of average rank of Gaussian distribution learners using IBL.The the x-axis refers to the learners in increasing order is - kdeKS.gaus, kdeNP.gaus, kde.gaus, kdeKD.gaus, kdeSM.gaus, dens.kde.gaus.ibl, dens.kde.gaus.ll, dens.kde.gaus.psl. The y-axis is the average rank obtained by ranking the mean IBL of each learners for each dataset and averaged the rank over all datasets for each learners.

Friedman test indicates there is a significant difference between the learners with p-value of 0.00296 ($\chi^2_7 = 21.6$). The critical difference (CD) from Nemenyi test is 1.950. The CD diagram is shown in Figure 8.9. The horizotal lines show groups of learners in which the difference in their ranks do not exceed the CD, indicating they are not significant different from each other.

From Figure 8.9, kdeNP.gaus is ranked better than other learners including tuned learners. Furthermore, dens.kde.gaus.ibl is ranked higher than other tuned learners.

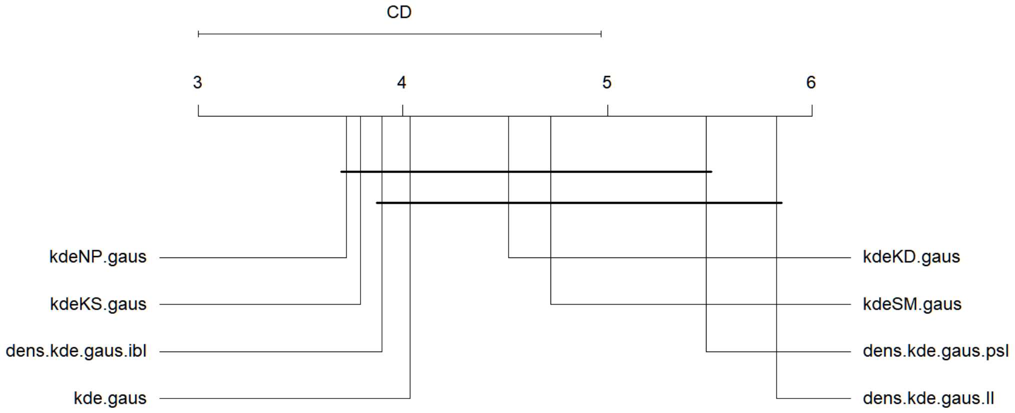


Figure 8.9: Critical difference (CD) diagram from the result of Nemenyi post-hoc test for using IBL as the evaluation method. The learners are ranked increasingly. The learners with the lowest rank is better than those with a higher rank. The critical distance (CD) is 1.950. The horizontal lines show groups of learners that are not significantly different.

8.4 Discussion & Conclusion

In this benchmarking experiment, we compare multiple distribution learners on multiple datasets and used different loss functions to evaluate. The learners are collected in **mlr3proba** and **mlr3extralearners**. However, there are limitation of this experiment, since we focus on the learners implement in **R** and has the functionality to estimate PDF or CDF at independent points. Therefore, we cannot ensure the that test data is used to estimate the distribution during evaluation. In evaluating, not all learners are being compared using PSL and IBL due to the limitation of computation.

From this experiment, different loss functions provide different conclusion. We summarize the result based on our objectives as below.

i. Overall, kdeKS.gaus has better performance when compared using log-loss, PSL and IBL. Although it did not rank first, it still ranked 2 or 3 from the Friedman-test.

ii. Using log-loss to compare the distribution learners, kdeKD.gaus has the minimum average log-loss. However, in terms of average rank and critical difference plot in Figure 8.2, dens.knn.sil and dens.logspline are ranked 1 and 2, respectively. kdeKS.gaus is rank third.

iii. Based on comparing the family of learners using log-loss, plug-in learners rank first in the *average loss* and has the minimum *average log-loss*. Meanwhile, tuned kernel learners has the highest average log-loss of 2.653 and rank last with *average rank*.

iv. Using PSL to compare kernel learners, kdeKS.gaus has the minimum *average PSL* and ranked first in the critical difference plot in Figure 8.4 while kdeKD.gaus is first in terms of average rank.

v. Using IBL to compare the learners, dens.kde.gaus.ll has the minimum *average IBL* while kdeKS.gaus is first in terms of average rank. However, kdeNP.gaus is ranked first in the critical difference plot in Figure 8.9.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

In this thesis, we explored and investigated distribution estimation task in machine learning.

Chapter 3 frames distribution estimation as supervised learning. The task is to learn a function that estimate a distribution using an unpaired dataset. The loss functions evaluates the estimated distribution at a point. The loss functions are indeed proper. Furthermore, the divergence between the generalization loss of the estimated distribution and the generalization loss of the true distribution leads to expected Kullback-Leibler divergence and mean integrated squared error.

Chapter 5 proposed an efficient method to compute the analytic expression of the probabilistic loss functions (log-loss, PSL, IBL) to evaluate the loss given a kernel-based distribution and an observation point. The proposed method is efficient because it not only able to compute and evaluate the loss of one kernel but is applicable for most kernel distribution and can be extended to kernel mixture distributions. From this method, closed-form expression of CDF, L2-norm PDF, L2-norm CDF and L2-norm CCDF of 11 symmetric kernel functions are derived which can be substituted into this method. Algorithms to compute the loss for mixtures are also provided.

Chapter 6 investigates the behaviour between in-sample and out-of-sample tuning in bandwidth selection using a Gaussian kernel PDF estimator and log-loss and PSL. From this investigation, out-of-sample tuning using log-loss and Gaussian kernel requires one new and unseen data points in the test data. Meanwhile, in-sample tuning is unbounded. For out-of-sample tuning using PSL and Gaussian kernel, the test set requires a ration such that, the total number of test data points to the number of observed data points in training and test sets is $2\sqrt{2} : 1$.

Chapter 7 provides a unified machine learning interface for distribution estimation in **R** which is integrated under **mlr3proba**. This interface is provides an easy and consistent implementation of distribution estimation. Using this platform, user can train, predict and evaluates distribution estimation. In addition, because **mlr3proba** lies within the **mlr3** ecosystem, user can use the functionalities offered by the ecosystem. This mean, user can also tune the parameter of a distribution learner and also perform benchmarking exercise.

Chapter 8 is a benchmarking experiment that compares multiple distribution on multiple datasets. In this experiment different distribution learners that includes tuned kernel learners, plug-in kernel learners, histogram learners, KNN and penalized distribution learners are compared on the 54 datasets using three different loss functions. The results indicate that plug-in kernel learners have a better performance compared to others.

9.2 Future Work

9.2.1 Investigating Tuning for Multivariate Kernel Estimators

A possible future work is to extend the investigation of tuning parameters for multivariate distribution. Based on Chapter 6, it is important that for out-of-sample tuning to work in selection of optimal bandwidth, there must be test data points that are distinct from the training data points. It would be interesting to investigate whether this condition applies to multivariate distribution estimation, particularly when using PSL for tuning. There are different questions that can arise from this investigation. Firstly, whether a higher number of dimension affects the ratio of distinct test to training data points when using PSL. Secondly, whether dependence or independence between covariates affect this condition.

9.2.2 Investigation of Ensemble Learning for Distribution Estimation

One possible future work is to compare all meta-learning methods (tuning, bagging, boosting and stacking) for distribution estimation. Ensemble learning for distribution estimation has been studied by [79], [82], [153], [48] [80] and [71]. [79] showed that bagging indeed reduce variance and hence reduce the MISE. [71] used log-loss to evaluate different meta-learning methods on distribution estimation and found that stacking distribution estimation showed a better result. [79], [82], [48] and [80] compared different distribution algorithms (including meta-learning) using empirical MISE and simulated dataset. However, when using real world data, it is possible that repeated data points are observed in both training and test sets. One possible future research is comparing how much bias and variance are reduced when using different meta-learning methods mentioned. Another possible investigation is how does different meta-learning algorithms affects multivariate distribution estimation.

9.2.3 Extending mlr3proba

The development of this interface for distribution estimation is still maturing. There are things that can be improved, not by the design but in terms of functionality. Firstly, **mlr3proba**, **weka** and **scikit-learn** use the same scoring methods (log-loss) to evaluate distribution estimation. However, there are also PSL and IBL that can be added, this is especially for kernel methods since **distr6** has the methods to compute the terms needed for PSL and IBL. To include PSL and IBL for kernel methods into **mlr3proba**, we need to consider: (1) improve speed of computation; (2) alternative for IBL for Normal and Sigmoid kernels in **distr6**. For improving the computation speed, it is more efficient to use Rccp as [7] used for `dpqr` methods. An alternative for IBL of Normal and Sigmoid kernel is by using quadrature rule instead of integration of L2-norm of CDF. Secondly, extending the univariate distribution interface to multivariate distribution interface is future work that can be looked into as some of the distribution learners in `R` are compatible for multivariate distribution estimation, such as the learners in `ks`, `np` and `sm` packages. Third, adding ensemble learning methods (e.g. bagging, stacking and etc) for distribution estimation which is not common in `R`. Several papers ([182], [71]) studied ensemble methods for distribution estimation. The algorithms used are not straightforward and will allow users to quickly implement the learners.

Appendices

Appendix A

Non-parametric Kernel Distribution Estimation

The derivations are based on [55] for references purpose.

A.1 Expectation of kernel density estimator

Let $X_1, \ldots, X_N \overset{i.i.d}{\sim} X$ and X t.v.i $\mathbb{R}$. Let $\hat{f}$ be a kernel density estimator (KDE) as in eqn (2.3.4) with bandwidth h. Let $K(u)$ be a kernel function that satisfies the following

$$\begin{aligned}\int K(u)\;du =&1\\ \int uK(u)\;du =&0\\ \int u^2K(u)\;du =&k_2 > 0.\end{aligned}$$

The kernel PDF estimator is

$$\hat{f}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x-X_i}{h}\right).$$

The expectation of $\hat{f}(x)$ is

$$\begin{aligned}\mathbb{E}[\hat{f}(x)] =&\frac{1}{Nh}\sum_{i=1}^{N}\mathbb{E}\left[K\left(\frac{x-X_i}{h}\right)\right] && \text{(A.1.1)}\\ =&\mathbb{E}\left[K\left(\frac{x-X}{h}\right)\right] && \text{(A.1.2)}\\ =&\int\frac{1}{h}K\left(\frac{x-t}{h}\right)f(t)\;dt && \text{(A.1.3)}\end{aligned}$$

By change of variable, such that $w = \frac{x-t}{h} \rightarrow t = x - uh$. Then, differentiate w w..r.t t, will obtained $dw = -h\;dt$ and

$$\mathbb{E}[\hat{f}(x)] \int K(w)f(x-wh)\;dw \qquad \text{(A.1.4)}$$

$$\text{(A.1.5)}$$

and using Taylor expansion on $f(x-wh)$,

$$\begin{aligned}f(x-wh) =&f(x) + (-uh)f'(x) + \frac{(-(uh)^2)}{2!}f\text{"}(x) + ... && \text{(A.1.6)}\\ =&f(x) - uhf'(x) + \frac{(uh)^2}{2!}f''(x) + ... && \text{(A.1.7)}\end{aligned}$$

Then, the expection of $\hat{f}(x)$ is

$$\mathbb{E}[\hat{f}(x)] = \int K(u) \left[f(x) + (-uh)f'(x) + \frac{(-(uh)^2)}{2!} f"(x) + ... \right] du \tag{A.1.8}$$

$$= f(x) \int K(u)du - hf'(x) \int uK(u)du + \frac{(h)^2}{2!} f''(x) \int u^2 K(u)du + \mathcal{O}(h^4) \tag{A.1.9}$$

By the properties of kernel function in section 5.3,

$$\mathbb{E}[\hat{f}(x)] = f(x) + \frac{h^2 \kappa_2}{2} f''(x) + \mathcal{O}(h^4) \tag{A.1.10}$$

Then, we can show that

$$\text{Bias}[\hat{f}(x)] = f(x) + \frac{h^2 \kappa_2}{2} f''(x) + \mathcal{O}(h^2) - f(x) \tag{A.1.11}$$

$$= \frac{h^2 \kappa_2}{2} f''(x) + \mathcal{O}(h^4) \tag{A.1.12}$$

A.2 Variance of kernel density estimator

$$\text{Var}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)^2] - \mathbb{E}[\hat{f}(x)]^2 \tag{A.2.1}$$

Then,

$$\mathbb{E}[\hat{f}(x)^2] = \frac{1}{N^2 h^2} \sum_{i=1}^{N} \mathbb{E} \left[K \left(\frac{x - X_i}{h} \right)^2 \right] \tag{A.2.2}$$

$$= \frac{1}{Nh^2} \mathbb{E} \left[K \left(\frac{x - X}{h} \right)^2 \right] \tag{A.2.3}$$

$$= \frac{1}{Nh^2} \int_{-\infty}^{\infty} K \left(\frac{x - t}{h} \right)^2 f(t)\, dt \tag{A.2.4}$$

Then, applying the substitution $w = \frac{x-t}{h} \to t = x - wh$ and the derivatives $\frac{dt}{dw} = -h \to dt = -h\, dw$.

$$\mathbb{E}[\hat{f}(x)^2] = \frac{1}{Nh^2} \int K(w)^2 f(x - wh) - hdw \tag{A.2.5}$$

$$= \frac{1}{Nh} \int K(w)^2 f(x - wh) dw \tag{A.2.6}$$

Then, use Taylor's expansion for $x - wh = 0$,

$$f(x - wh) = f(x) + (-wh)f'(x) + \frac{(-wh)^2}{2}f''(x) + ... \tag{A.2.7}$$

and substituting the expansion back into eqn (A.2.5) to obtain

$$\mathbb{E}[\hat{f}(x)^2] = \frac{1}{Nh}\left(\int K(w)^2[f(x) + (-wh)f'(x) + \frac{(-wh)^2}{2}f''(x) + ...]\right) dw \tag{A.2.8}$$

$$= \frac{1}{Nh}\left(f(x)\int K(w)^2 dw - hf'(x)\int wK(w)^2\, dw + \right.$$
$$\left.\frac{h^2 f''(x)}{2}\int w^2 K(w)^2\, dw\right) \tag{A.2.9}$$

by the properties of kernel function, $K(u)$,

$$\mathbb{E}[\hat{f}(x)^2] = \frac{1}{Nh}\left[f(x)\int K(w)^2 dw + \frac{(wh)^2}{2}f''(x)\int w^2K(w)^2 dw + \mathcal{O}(h^2)\right] \tag{A.2.10}$$

Then, using eqn (A.2.10) and the mean of $\hat{f}(x)$, the variance is

$$Var[\hat{f}(x)] = \frac{1}{Nh}\left[f(x)\int K(w)^2 dw + \frac{(wh)^2}{2}f''(x)\int w^2K(w)^2 dw + \mathcal{O}(h^2)\right] - \tag{A.2.11}$$

$$[f(x) + \frac{h^2k^2}{2}f''(x) + \mathcal{O}(h^2)]^2 \tag{A.2.12}$$

$$= \frac{1}{Nh}f(x)\int K(w)^2 dw + \mathcal{O}\left(\frac{h}{N}\right) \tag{A.2.13}$$

Appendix B

Chapter 5: Efficient Computation of Loss Functions for Distribution Estimation

B.1 Derivation for PDF, CDF, partial L2-products of PDF, CDF and CCDF

In this section of appendix are the derivations of the functions in Table 5.1.

B.2 Uniform Kernel

Definition B.2.1. *Let the Uniform kernel be defined as*

$$K(u) = \begin{cases} \frac{1}{2} & \textit{if} \quad |u| \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

Derivation B.2.1. *Let K be a Uniform kernel as in Definition B.2.1. Then, the CDF for uniform kernel is*

$$\nu_K(t) = \begin{cases} 0 & \textit{if} \quad t \leq -1 \\ \frac{1}{2}(t+1) & \textit{if} \quad -1 \leq t \leq 1 \\ 1 & \textit{if} \quad t \geq 1 \end{cases}$$

Proof. Let K be a uniform kernel from Definition B.2.1. The CDF is the integration of the kernel function. There are three cases to consider: (1) $t \leq -1$; (2) $-1 \leq t \leq 1$; (3) $t \geq 1$.

i. **Case 1** $t \leq -1$**:** Under this case, the integration of K is outside the range of Definition B.2.1 resulting $\int_{-\infty}^{t} K(u)\, du = 0$.

ii. **Case 2** $t \in [-1, 1]$**:**

$$\int_{-\infty}^{t} \frac{1}{2}\mathbb{1}(u \in [-1,1])\, du = \left[\frac{1}{2}u\right]_{-1}^{t} = \frac{1}{2}(t+1) \tag{B.2.1}$$

iii. **Case 3** $t \geq 1$**:**

$$\int_{-\infty}^{t} \frac{1}{2}\mathbb{1}(u \in [-1,1])\, du = 1 \tag{B.2.2}$$

□

Derivation B.2.2. *Let K be a uniform kernel as in Definition B.2.1. Then, the integration of two Uniform kernel at two different centres, 0 and $c \in \mathbb{R}$, from $[-\infty, \infty]$ is*

$$\lambda_K(a = \infty, c) = \begin{cases} \frac{1}{4}(2 - |c|) & \textit{if} \quad |c| \leq 2 \\ 0 & \textit{if} \quad |c| > 2. \end{cases}$$

Proof. Suppose that we have the uniform kernel as in Definition B.2.1. Then, the partial L2-product of the kernel as in Eqn (5.4.2) for uniform kernel is

$$\begin{aligned}\lambda_K(c) =& \int_{-\infty}^{a=\infty} K(u)K(u-c)\mathbb{1}(\in[-1,1])\mathbb{1}(u\in[c-1,c+1])\,du\\ =&\frac{1}{4}\int_{-\infty}^{\infty}\mathbb{1}(u\in[-1,1])\mathbb{1}(u\in[c-1,c+1])\,du. \end{aligned}\tag{B.2.3}$$

There two cases to consider in this computation: (1) $|c|\le 2$; (2) $|c|\ge 2$. For (2), there is no intersection between the two kernels resulting $\lambda_K(a=\infty,c)=0$. Hence, the computation only focusses on (1). Under (1), there are two sub-cases to consider:

i. Case 1(a) $c\in[c-1,1]$:

$$\lambda_K(c)=\int_{c-1}^{1}\frac{1}{4}\,du=\frac{1}{4}(2-c)\tag{B.2.4}$$

ii. Case 1(b) $c\in[-1,c+1]$: This is another case where the intersection occurs between the two kernels, but between $[-1,c+1]$.

$$\lambda_K(c)=\int_{-1}^{c+1}\frac{1}{4}\,du=\frac{1}{4}(2+c)\tag{B.2.5}$$

□

Derivation B.2.3. *Suppose we have the Uniform kernel as in Definition B.2.1. Then, the integration of two Uniform kernel with to different starting points* 0 *and* $c\in\mathbb{R}$ *from* $[-\infty,a]$ *where* $a\in\mathbb{R}$ *is*

i. $c\in[0,2]$

$$\lambda_K(a,c)=\begin{cases}\frac{1}{4}(2-c) & \textit{if} \quad a\ge 1\\ \frac{1}{4}(a-c+1) & \textit{if} \quad a\in[c-1,1]\\ 0 & \textit{if} \quad a\le c-1\end{cases}$$

ii. For $c\in[-2,0]$

$$\lambda_K(a,c)=\begin{cases}\frac{1}{4}(2+c) & \textit{if} \quad a\ge c+1\\ \frac{1}{4}(a+1) & \textit{if} \quad a\in[-1,c+1]\\ 0 & \textit{if} \quad a\le -1.\end{cases}$$

Proof. Suppose that we have the uniform kernel as in Definition B.2.1. Then, the partial L2-product of the kernel as in Eqn (5.4.2) for uniform kernel is

$$\begin{aligned}\lambda_K(a,c) =& \int_{-\infty}^{a} K(u)K(u-c)\mathbb{1}(\in[-1,1])\mathbb{1}(u\in[c-1,c+1])\,du \\ =&\frac{1}{4}\int_{-\infty}^{a} \mathbb{1}(u\in[-1,1])\mathbb{1}(u\in[c-1,c+1])\,du. \end{aligned} \tag{B.2.6}$$

There 3 cases that we need to consider: (1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $|c| > 2$. For (3) this is outside the intersection of the two Uniform kernels. Hence, resulting $\lambda_K(a,c) = 0$. Hence, the computation of $\lambda_K(a,c)$ for uniform kernel only focuses on two case (1) and (2).

i. **Case 1** $c \in [0,2]$**:** The intersection occurs between $[c-1,1]$. Under this condition, we need to consider three subcases below:

a. Case 1(a) $a \geq 1$:

$$\lambda_K(a,c) = \int_{c-1}^{1} \frac{1}{4}\,du = \frac{1}{4}\left(1-(c-1)\right) = \frac{1}{4}(2-c)$$

b. Case 1(b) $c-1 \leq a \leq 1$:

$$\lambda_K(a,c) = \int_{c-1}^{a} \frac{1}{4}\,du = \frac{1}{4}\left(a-c+1\right) \tag{B.2.7}$$

c. Case 1(c) $a \leq c-1$:

$$\lambda_K(a,c) = \int_{-\infty}^{a} \frac{1}{4}\,du = 0 \tag{B.2.8}$$

ii. **Case 2** $c \in [-2,0]$: The intersection occurs between $[-1,c+1]$. Under this condition, we need to consider three subcases below:

a. Case 2(a) $a > c+1$:

$$\lambda_K(a,c) = \int_{-1}^{c+1} \frac{1}{4}\,du = \frac{1}{4}\left(2+c\right) \tag{B.2.9}$$

b. Case 2(b) $-1 \leq a \leq c+1$:

$$\lambda_K(a,c) = \int_{-1}^{a} \frac{1}{4}\,du = \frac{1}{4}\left(a+1\right) \tag{B.2.10}$$

c. Case 2(c) $a < -1$:

$$\lambda_K(a,c) = \int_{-\infty}^{a} \frac{1}{4}\, du = 0 \tag{B.2.11}$$

□

Derivation B.2.4. *Let K be a Uniform kernel defined as in Definition (B.2.1). The L2 norm of uniform CDF two different central points 0 and $c \in \mathbb{R}$ is*

i. For $c \in [0,2]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq -1 \\ 0 & \textit{if} \quad a \in [-1, c-1] \\ \frac{c^3+2a^3+3a^2(2-c)+6a(1-c)+2-3c}{24} & \textit{if} \quad a \in [c-1, 1] \\ \frac{c^3+6a^2+12a-12ac-2}{24} & \textit{if} \quad a \in [1, c+1] \\ \frac{c^3-6c^2-12c+24a-8}{24} & \textit{if} \quad a \geq c+1 \end{cases}$$

ii. For $c \in [-2,0]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq c-1 \\ 0 & \textit{if} \quad a \in [c-1, -1] \\ \frac{2a^3+3a^2(2-c)+6a(1-c)+2-3c}{24} & \textit{if} \quad a \in [-1, c+1] \\ \frac{-c^3+6(a^2-c^2)+12(a-c)-2}{24} & \textit{if} \quad a \in [c+1, 1] \\ \frac{-c^3-6c^2-12c+24a-8}{24} & \textit{if} \quad a \geq 1 \end{cases}$$

iii. For $c \geq 2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a < -1 \\ 0 & \textit{if} \quad a \in [-1, 1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ \frac{c^2-2c+1}{4} + \frac{a(2-2c)}{4} + \frac{a^2}{4} & \textit{if} \quad a \in [c-1, c+1] \\ a-c & \textit{if} \quad a \geq c+1 \end{cases}$$

iv. For $c \leq -2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq c-1 \\ 0 & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \in [c+1, -1] \\ \frac{(a+1)^2}{4} & \textit{if} \quad a \in [-1, 1] \\ a & \textit{if} \quad a \geq 1 \end{cases}$$

Proof. Let K be a Uniform kernel as in Def B.2.1. The CDF is defined in B.2.1. Then, the partial L2-product of CDF as in Eqn (5.4.3) is

$$\gamma_K(a,c) = \int_{-\infty}^{a} \frac{1}{4}(u+1)(u-c+1)\mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [c-1, c+1])\, du. \tag{B.2.12}$$

There are several things to consider in computing $\gamma_K(a,c)$ for Uniform kernel. For $c \geq 0$, the intersection of two uniform CDF's will occur from $c-1$ until ∞, whereas for $c \leq 0$ the intersection occurs between $[0, \infty)$. Therefore, we need to consider these cases: (1) $c \in [0,2]$; (2) $[-2,0]$; (3) $c \geq 2$; (4) $c \leq -2$.

i. **Case 1:** $c \in [0,2]$**:** Under this case, we need to consider 4 subcases below:

a. Case 1(a): $a \leq c-1$: In this case, there is no overlapping between $\nu(t)$ and $\nu_K(t-c)$, hence

$$\gamma_K(a,c) = \int_{-1}^{c-1} \frac{1}{4}(t+1)(t-c+1)\, dt = 0 \tag{B.2.13}$$

b. Case 1(b) $a \in [c-1, 1]$: The partial L2-product of uniform cdf for the overlapping between $\nu(t)$ and $\nu(t-c)$ is

$$\begin{aligned}\gamma_K(a,c) &= \int_{c-1}^{a} \frac{1}{4}(t+1)(t-c+1)\, dt \\ &= \frac{c^3 + \left(-3a^2 - 6a - 3\right)c + 2a^3 + 6a^2 + 6a + 2}{24}\end{aligned} \tag{B.2.14}$$

When $a = c-1$,

$$\frac{c^3 + \left(-3a^2 - 6a - 3\right)c + 2a^3 + 6a^2 + 6a + 2}{24} = 0 \tag{B.2.15}$$

When $a = 1$,

$$\frac{c^3 + (-3a^2 - 6a - 3)\,c + 2a^3 + 6a^2 + 6a + 2}{24} = \frac{1}{24}\left(c^3 - 12c + 16\right)$$

c. Case 1(c) $a \in [1, c+1]$:

$$\gamma_K(a,c) = \frac{1}{4}\int_{c-1}^{1} (t+1)(t-c+1)\;dt + \int_{1}^{a} \frac{1}{2}(t-c+1)\;dt \tag{B.2.16}$$

$$= \frac{c^3 + 6a^2 + 12a - 12ac - 2}{24} \tag{B.2.17}$$

When $a = 1$,

$$\frac{c^3 + 6a^2 + 12a(1-c) - 2}{24} = \frac{1}{24}\left(c^3 - 12c + 16\right) \tag{B.2.18}$$

When $a = c + 1$,

$$\frac{c^3 + 6a^2 + 12a(1-c) - 2}{24} = \frac{1}{24}\left(c^3 - 6c^2 + 12c + 16\right) \tag{B.2.19}$$

d. Case 1(d) $a \geq c + 1$:

$$\gamma_K(a,c) = \frac{1}{4}\int_{c-1}^{1} (t+1)(t-c+1)\;dt + \int_{1}^{c+1} \frac{1}{2}(t-c+1)\;dt + \int_{c+1}^{1} 1\;dt \tag{B.2.20}$$

$$= \frac{c^3 - 12c + 16}{24} - \frac{(c-4)\,c}{a} + (a - (c+1)) \tag{B.2.21}$$

$$= \frac{c^3 - 6c^2 - 12c + 24a - 8}{24} \tag{B.2.22}$$

When $a = c + 1$,

$$\frac{c^3 - 6c^2 - 12c + 24a - 8}{24} = \frac{1}{24}\left(c^3 - 6c^2 + 12c + 16\right) \tag{B.2.23}$$

ii. **Case 2** $c \in [-2, 0]$**:** Under this condition, we will need to consider 4 sub-cases below:

a. Case 2(a) $a \leq -1$:

$$\gamma_K(a,c) = \int_{-1}^{c-1} \frac{1}{4}(t+1)(t-c+1)\;dt = 0 \tag{B.2.24}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\gamma_K(a,c) = \int_{-1}^{a} \frac{1}{4}(t+1)(t-c+1)\, dt \tag{B.2.25}$$

$$= \frac{1}{24}(2a^3 + 3a^2(2-c) + 6a(1-c) + 2 - 3c) \tag{B.2.26}$$

Check: When $a = -1$,

$$\frac{1}{24}(2a^3 + 3a^2(2-c) + 6a(1-c) + 2 - 3c) = 0 \tag{B.2.27}$$

Check: When $a = c+1$,

$$\frac{1}{24}(2a^3 + 3a^2(2-c) + 6a(1-c) + 2 - 3 = \frac{1}{24}\left(-c^3 + 12c + 16\right) \tag{B.2.28}$$

c. Case 2(c) $a \in [c+1, 1]$:

$$\gamma_K(a,c) = \int_{-1}^{c+1} \frac{1}{4}(t+1)(t-c+1)\, dt + \int_{c+1}^{a} \frac{1}{2}(1+t)\, dt \tag{B.2.29}$$

$$= \frac{-c^3 + 12c + 16}{24} + \frac{1}{4}(a^2 + 2a - c^2 - 4c - 3) \tag{B.2.30}$$

$$= \frac{-c^3 - 6c^2 - 12c - 2 + 6a^2 + 12a}{24} \tag{B.2.31}$$

Check: When $a = c+1$

$$\frac{-c^3 - 6c^2 - 12c - 2 + 6a^2 + 12a}{24} = \frac{-c^3 + 12c + 16}{24} \tag{B.2.32}$$

Check: When $a = 1$

$$\frac{-c^3 - 6c^2 - 12c - 2 + 6a^2 + 12a}{24} = \frac{1}{24}\left(-c^3 - 6c^2 - 12c + 16\right) \tag{B.2.33}$$

d. Case 2(d): $a \geq 1$

$$\gamma_K(a,c) = \int_{-1}^{c+1} \frac{1}{4}(t+1)(t-c+1)\, dt + \int_{c+1}^{1} \frac{1}{2}(1+t)\, dt + \int_{1}^{a} 1\, dt \tag{B.2.34}$$

$$= \frac{-c^3 + 12c + 16}{24} - \frac{c\,(c+4)}{4} + (a-1) \tag{B.2.35}$$

$$= \frac{-c^3 - 6c^2 - 12c + 24a - 8}{24} \tag{B.2.36}$$

Check: When $a = 1$

$$\frac{-c^3 - 6c^2 - 12c + 24a - 8}{24} = \frac{1}{24}\left(-c^3 - 6c^2 - 12c + 16\right) \qquad \text{(B.2.37)}$$

iii. **Case 3** $c \geq 2$**:** Under this condition, the intersection of the two CDF's $\nu(t)$ and $\nu(t - c)$ can occur when:

a. Case 3(a) $a \in [c - 1, 1]$:

$$\gamma_K(a, c) = \int_{c-1}^{a} \frac{t - c + 1}{2}\, dt = \frac{c^2 - 2c + 1}{4} + \frac{a\,(2 - 2c)}{4} + \frac{a^2}{4} \qquad \text{(B.2.38)}$$

b. Case 3(b) $a \geq 1$:

$$\gamma_K(a, c) = \int_{c-1}^{c+1} \frac{t - c + 1}{2}\, dt + \int_{c+1}^{a} 1 dt = a - c \qquad \text{(B.2.39)}$$

iv. **Case 4** $c \leq -2$**:** Under this condition, the intersection of the two cdf's $\nu(t)$ and $\nu(t - c)$ can occur when:

a. Case 4(a) $a \in [-1, +1]$:

$$\gamma_K(a, c) = \int_{-1}^{a} \frac{t + 1}{2}\, dt = \frac{(a + 1)^2}{4} \qquad \text{(B.2.40)}$$

b. Case 4(b) $a \geq 1$:

$$\gamma_K(a, c) = \int_{c-1}^{c+1} \frac{t - c + 1}{2}\, dt + \int_{c+1}^{a} 1 dt = a \qquad \text{(B.2.41)}$$

□

Derivation B.2.5. *Let K be a Uniform kernel function with CDF ν. The partial L2-product of the CCDF $(1 - \nu)^2$ at two different central points 0 and $c \in \mathbb{R}$ as in Eqn (5.4.4) is*

i. For $c \in [0, 2]$

$$\xi_K(a, c) = \begin{cases} \frac{c^3 - 6c^2 + 12c - 24a - 8}{24} & \text{if} \quad a \leq -1 \\ \frac{c^3 + 6(a^2 - c^2) + 12(c - a) - 2}{24} & \text{if} \quad a \leq [-1, c - 1] \\ \frac{-2a^3 + 3a^2(2 + c) - 6a(1 + c) + 3c + 2}{24} & \text{if} \quad a \in [c - 1, 1] \\ 0 & \text{if} \quad a \in [1, c + 1] \\ 0 & \text{if} \quad a \geq c + 1 \end{cases}$$

ii. For $c \in [-2, 0]$

$$\xi_K(a,c) = \begin{cases} \frac{-c^3-6c^2+12c-24a-8}{24} & \textit{if} \quad a \leq c-1 \\ \frac{-c^3+6a^2-12ac-12a-2}{4} & \textit{if} \quad a \leq [c-1, -1] \\ \frac{-2a^3+3a^2(c+2)-6a(c+1)-c^3+3c+2}{24} & \textit{if} \quad a \in [c-1, 1] \\ 0 & \textit{if} \quad a \in [1, c+1] \\ 0 & \textit{if} \quad a \geq c+1 \end{cases}$$

iii. For $c \geq 2$

$$\xi_K(a,c) = \begin{cases} -a & \textit{if} \quad a < -1 \\ \frac{(1-a)^2}{4} & \textit{if} \quad a \in [-1, 1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ 0 & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \geq c+1 \end{cases}$$

iv. For $c \leq 2$

$$\xi_K(a,c) = \begin{cases} c-a & \textit{if} \quad a \leq c-1 \\ \frac{c^2+2c+1}{4} + \frac{a(-2c-2)}{4} + \frac{a^2}{4} & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \in [c+1, -1] \\ 0 & \textit{if} \quad a \in [-1, 1] \\ 0 & \textit{if} \quad a \geq 1 \end{cases}$$

Proof. Form Lemma 5.4.2, $\xi_K(a,c)$ is the reflection of $\gamma_K(a,c)$ on the y-axis. Hence, we can compute $\xi_K(a,c)$ by taking $\gamma_K(-a,-c)$. □

B.3 Epanechnikov Kernel

Definition B.3.1. *Let the Epanechnikov Kernel be defined as*

$$K(u) = \begin{cases} \frac{3}{4}(1-u^2) & \textit{if} \quad |u| \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

Derivation B.3.1. *Let K be an Epanechnikov kernel as defined in Def B.3.1. Then, the CDF of the Epanechnikov kernel is*

$$\nu_K(t) = \begin{cases} 0 & \textit{if} \quad t \leq -1 \\ \frac{-t^3+3t+2}{4} & \textit{if} \quad t \in [-1,1] \\ 1 & \textit{if} \quad t \geq 1 \end{cases}$$

Proof. Let K be a Epanechnikov kernel as in Def B.3.1. The cdf is the integration of the kernel function. There are three cases to consider: (1) $t \leq -1$; (2) $t \in [-1,1]$; (3) $t \geq 1$.

i. **Case 1** $a \leq -1$**:** Under this case, the integration of K is outside the range of Definition B.3.1 resulting $\int_{-\infty}^{t} K(u)\, du = 0$.

ii. **Case 2** $t \in [-1,1]$**:**

$$\nu_K(t) = \int_{-1}^{t} \frac{3}{4}(1-u^2)\, du = \frac{-t^3+3t+2}{4} \tag{B.3.1}$$

iii. **Case 3** $t \geq 1$**:**

$$\nu_K(t) = \int_{-1}^{1} \frac{3}{4}(1-u^2)\, du = 1 \tag{B.3.2}$$

□

Derivation B.3.2. *Let K be an Epanechnikov kernel as in Def B.3.1. Then, the partial L2-product of Epanechnikov kernels at two different central point 0 and $c \in \mathbb{R}$ when the integration boundary is $(-\infty, \infty)$ is*

$$\lambda_K(c) = \begin{cases} -\frac{|c|^5-20|c|^3+40|c|^2-32}{30} & \textit{if} \quad |c| \leq 2 \\ 0 & \textit{if} \quad |c| > 2 \end{cases}$$

Proof. Suppose that we have the Epanechnikov kernel as in Definition B.3.1. the partial L2-product of the kernel as in Eqn (5.4.2) for Epanechnikov kernel is

$$\lambda_K(c) = \int (1-u^2)(1-(u-c)^2)\mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [c-1,c+1])\, du$$

There are three cases that we need to consider in this computation: (1) $|C \leq 2$; (2) $|c| \geq 2$. For (2), there is no intersection between the two kernels resulting in $\lambda_K(a=\infty, c) = 0$. Hence, the computation only focuses on (1).

i. Case 1(a) $c \in [c-1, 1]$:

$$\lambda_K(c) = \int_{c-1}^{1} (1-u^2)(1-(u-c)^2)\, du \tag{B.3.3}$$

$$= \left[u(1-c^2) + u^2c + \frac{u^3}{3}(c^2-2) - \frac{u^2c}{2} + \frac{u^5}{5} \right]_{c-1}^{1} \tag{B.3.4}$$

$$= -\frac{c^5 - 20c^3 + 40c^2 - 32}{30} \tag{B.3.5}$$

ii. Case 1(b) $c \in [-1, c+1]$:

$$\lambda_K(c) = \int_{c-1}^{1} (1-u^2)(1-(u-c)^2)\, du \tag{B.3.6}$$

$$= \left[u(1-c^2) + u^2c + \frac{u^3}{3}(c^2-2) - \frac{u^2c}{2} + \frac{u^5}{5} \right]_{-1}^{c+1} \tag{B.3.7}$$

$$= \frac{c^5 - 20c^3 - 40c^2 + 32}{30}. \tag{B.3.8}$$

□

Derivation B.3.3. *Let K be an Epanechnikov kernel as in Def B.3.1. Then, the the partial L2-product of Epanechnikov kernels at two different central point 0 and $c \in \mathbb{R}$ when the integration boundary is $(-\infty, a]$ where $a \in \mathbb{R}$ is*

i. ***For*** $c \in [0, 2]$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{3(-c^5+20c^3+10c^2\left(a^3-3a-2\right)-15c(a^2-1)^2)}{160} + & \\ \frac{3(6a^5-20a^3+30a+16)}{160} & \textit{if } a \in [c-1, 1] \\ \frac{3(-c^5+20c^3-40c^2+32)}{160} & \textit{if } a \geq 1 \end{cases}$$

ii. ***For*** $c \in [-2, 0]$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{3(10c^2\left(a^3-3a-2\right)-15c(a^2-1)^2+6a^5-20a^3+30a+16)}{160} & \textit{if } a \in [-1, c+1] \\ \frac{3(c^5-20c^3-40c^2+32)}{160} & \textit{if } a \geq c+1 \end{cases}$$

Proof. Suppose that we have Epanechnikov kernel as in Definition B.3.1. Then, the L2 norm of the kernel as in Eqn (B.3.3) for Epanechnikov kernel is

$$\lambda_K(a,c) = \int_{-\infty}^{a} (1-u^2)(1-(u-c)^2)\mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [c-1, c+1])\, du \tag{B.3.9}$$

$$= \int_{-\infty}^{a} (1-u^2)(1-(u-c)^2)\mathbb{1}(u \in [-1,1] \cap [c-1, c+1])\, du \tag{B.3.10}$$

There are three cases that needed to be considered: (1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $|c| \geq 2$. For (3), the range is outside the intersection of the two Epanechnikov kernels resulting $\lambda_K(a,c) = 0$. Hence, the computation of $\lambda_K(a,c)$ for Epanechnikov kernel only focusses of cases (1) and (2).

i. **Case 1** $c \in [0,2]$**:** The intersection between the two lines happens between $[c-1, 1]$. Under this condition, we need to consider the cases below:

a. Case 1(a) $a \leq c-1$:

$$\lambda_K(a,c) = \frac{9}{16}\int_{c-1}^{c-1} (1-u^2)(1-(u-c)^2)du = 0 \tag{B.3.11}$$

b. Case 1(b): $a \in [c-1, 1]$

$$\begin{aligned}\lambda_K(a,c) =& \frac{9}{16}\int_{c-1}^{a} (1-u^2)(1-(u-c)^2)du \\ =& \frac{3(-c^5+20c^3+10c^2(a^3-3a-2)-15c(a^2-1)^2)}{160} + \\ & \frac{3(6a^5-20a^3+30a+16)}{160}\end{aligned} \tag{B.3.12}$$

Check: When $a = c - 1$

$$\begin{aligned}&\frac{3(-c^5+20c^3+10c^2\left(a^3-3a-2\right)-15c(a^2-1)^2+6a^5-20a^3+30a+16)}{160}\\=&0 \qquad \text{(B.3.13)}\end{aligned}$$

Check: When $a = 1$

$$\begin{aligned}&\frac{3(-c^5+20c^3+10c^2\left(a^3-3a-2\right)-15c(a^2-1)^2+6a^5-20a^3+30a+16)}{160}\\=&\frac{3(-c^5+20c^3-40c^2+32)}{160} \qquad \text{(B.3.14)}\end{aligned}$$

c. Case 2(c) $a \geq 1$:

$$\begin{aligned}\lambda_K(a,c)=&\frac{9}{16}\int_{c-1}^{1}(1-u^2)(1-(u-c)^2)du\\=&\frac{3}{160}\left(-c^5+20c^3-40c^2+32\right) \qquad \text{(B.3.15)}\end{aligned}$$

ii. **Case 2** $c \in [-2,0]$**:** The intersection occurs between $[-1, c+1]$. Under this condition, we need to consider the cases below:

a. Case 2(a) $a \leq -1$: In this case, a is outside the intersection region. Hence,

$$\lambda_K(a,c)=\frac{9}{16}\int_{-\infty}^{-1}(1-u^2)(1-(u-c)^2)du=0 \qquad \text{(B.3.16)}$$

b. Case 2(b) $a \in [-1, c+1]$: For this subcase, the limit of the integral is $[-1, a]$, hence

$$\begin{aligned}\lambda_K(a,c)=&\frac{9}{16}\int_{-1}^{a}(1-u^2)(1-(u-c)^2)du\\=&\frac{3(10c^2(a^3-3a-2)-15c(a^4-2a^2+1)+6a^5-20a^3+30a+16)}{160}\end{aligned} \qquad \text{(B.3.17)}$$

Check: When $a = -1$,

$$\begin{aligned}&\frac{3(10c(a^3-3a-2)-15c(a^4-2a^2+1)+6a^5-20a^3+30a+16)}{160}\\=&0 \qquad \text{(B.3.18)}\end{aligned}$$

Check: When $a = c + 1$

$$\begin{aligned} &\frac{3\left(10c^2(a^3-3a-2)-15c(a^4-2a^2+1)+6a^5-20a^3+30a+16\right)}{160} \\ =&\frac{3(c^5-20c^3-40c^2+32)}{160} \end{aligned} \tag{B.3.19}$$

c. Case 2(c) $a \geq c+1$: In this subcase, the limit of the integration is $[-1, c+1]$

$$\begin{aligned} \lambda_K(a,c) =& \frac{9}{16}\int_{-1}^{c+1}(1-u^2)(1-(u-c)^2)du \\ =&\frac{3\left(c^5-20c^3-40c^2+32\right)}{160} \end{aligned} \tag{B.3.20}$$

□

Derivation B.3.4. *Let K be an Epanechnikov kernel as defined in Def B.3.1. The partial L2-product of two CDF of Epanechnikov kernels at two different centre points 0 and $c \in \mathbb{R}$ from $-\infty$ to $a \in \mathbb{R}$ is*

i. For $c \in [0,2]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ 0 & \textit{if } \ a \in [-1, c-1] \\ \frac{a^7}{112} - \frac{a^6c}{32} + \frac{a^3(420+280c-420c^2)}{2240} + \frac{a^5(-168+84c^2)}{2240} + & \\ \frac{a^4(-140+420c-35c^3)}{2240} + \frac{a^2(840-630c-420c^2+210c^3)}{2240} + & \\ \frac{a(560-840c+280c^3)}{2240} + \frac{132-280c+84c^2+105c^3-42c^5+c^7}{2240} & \textit{if } a \in [c-1, 1] \\ \frac{c^7-42c^5+168c^2+560ac(c^2+a^2-3)+1120a+840a^2(1-c^2)-140a^4-156}{2240} & \textit{if } a \in [1, c+1] \\ \frac{c^7-42c^5+140c^4-672c^2-1120c+2240a-576}{2240} & \textit{if } a \geq c+1 \end{cases}$$

ii. For $c \in [-2,0]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ 0 & \textit{if } a \in [c-1, -1] \\ \frac{a\left(280c^3-840c+560\right)}{2240} + \frac{a^2\left(210c^3-420c^2-630c+840\right)}{2240} + & \\ \frac{105c^3+84c^2-280c+132}{2240} + \frac{a^4\left(-35c^3+420c-140\right)}{2240} + & \\ \frac{a^5\left(84c^2-168\right)}{2240} + \frac{a^3\left(-420c^2+280c+420\right)}{2240} - \frac{a^6c}{32} + \frac{a^7}{112} & \textit{if } a \in [-1, c+1] \\ \frac{-c^7+42c^5+140c^4-672c^2+1120(a-c)+840a^2-140a^4-156}{2240} & \textit{if } a \in [c+1, 1] \\ \frac{-c^7+42c^5+140c^4-672c^2-1120c+2240a-576}{2240} & \textit{if } a \geq 1 \end{cases}$$

iii. For $c \geq 2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq -1 \\ 0 & \textit{if} \quad a \in [-1,1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ \frac{-(-c+a-3)(-c+a+1)^3}{16} & \textit{if} \quad a \in [c-1, c+1] \\ a-c & \textit{if} \quad a \geq c+1 \end{cases}$$

iv. For $c \leq 2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq -1 \\ 0 & \textit{if} \quad a \in [-1,1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ \frac{-a^4+6a^2+8a+3}{16} & \textit{if} \quad a \in [c-1, c+1] \\ a & \textit{if} \quad a \geq c+1 \end{cases}$$

Proof. Let K be an Epanechnikov kernel as in Definition B.3.1 with CDF as in B.3.1. The partial L2-product of the CDF as in Eqn (**??**) is

$$\gamma_K(a,c) = \frac{1}{16}\int_{-\infty}^{a} (-t^3+3t+2)(-t^3+3ct^2+3t(1-c^2)+c^3-3c+2)\, dt$$

There are several cases needed to be considered in computing $\gamma_K(a,c)$ for Epanechnikov kernel. For $c \geq 0$, the overlapping of the two CDF happens when $[c-1,\infty)$. For $c \leq 0$, the overlapping of the two CDF occurs between $[0,\infty)$. Therefore, we need to consider these cases: (1) $c \in [0,2]$ (2) $[-2,0]$ (3) $c \geq 2$ (4) $c \leq -2$.

i. **Case 1** $c \in [0,2]$**:** Under this case, we need to consider 4 sub-cases below.

a. Case 1(a): $a \leq c-1$

$$\gamma_K(a,c) = 0 \tag{B.3.21}$$

b. Case 1(b): $a \in [c-1, 1]$

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{16}\int_{c-1}^{a}(-t^3+3t+2)(-(t-c)^3+3(t-c))\,dt \\ =& \frac{c^7-42c^5+105c^3+84c^2-280c+132}{2240}+\frac{a\left(280c^3-840c+560\right)}{2240}+ \\ & \frac{a^2\left(210c^3-420c^2-630c+840\right)}{2240}+\frac{a^4\left(-35c^3+420c-140\right)}{2240}+ \\ & \frac{a^5\left(84c^2-168\right)}{2240}+\frac{a^3\left(-420c^2+280c+420\right)}{2240}-\frac{a^6c}{32}+\frac{a^7}{112}\end{aligned} \tag{B.3.22}$$

Check: When $a = 1$,

$$\gamma_K(a,c) = \frac{c^7}{2240}-\frac{3c^5}{160}+\frac{c^3}{4}-\frac{3c^2}{10}-\frac{c}{2}+\frac{26}{35} \tag{B.3.23}$$

c. Case 1(c): $a \in [1, c+1]$

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{1}\frac{1}{16}(-t^3+3t+2)(-(t-c)^3+3(t-c)+2)\,dt+ \\ & \int_{1}^{a}\frac{1}{4}(-(t-c)^3+3(t-c)+2)\,dt \\ =& \frac{c^7}{2240}-\frac{3c^5}{160}+\frac{c^3}{4}-\frac{3c^2}{10}-\frac{c}{2}+\frac{26}{35}+ \\ & \frac{(a-1)\left(2c\left(c\left(2c-3a-3\right)+2\left(a-1\right)\left(a+2\right)\right)-a^3-a^2+5a+13\right)}{16} \\ =& \frac{c^7-42c^5+560ac^3-840a^2c^2+168c^2}{2240}+ \\ & \frac{560a^3c-1680ac-140a^4+840a^2+1120a-156}{2240}\end{aligned} \tag{B.3.24}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{85c^7-504c^6+798c^5-420c^4+560c^3-1792c+1664}{2240} \tag{B.3.25}$$

Check: When $a = c+1$

$$\frac{c^7-42c^5+140c^4-672c^2+1120c+1664}{2240} \tag{B.3.26}$$

d. Case 1(d): $a \geq c+1$

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{1} \frac{1}{16}(-t^3+3t+2)(-t^3+3ct^2+3t(1-c)+c^3-3c+2)\,dt + \\ & \int_{1}^{c+1} \frac{1}{4}(-t^3+3ct^2+3t(1-c)+c^3-3c+2)\,dt + \int_{c+1}^{a} 1\,dt \\ =& \frac{c^7-42c^5+140c^4-672c^2+1120c+1664}{2240} + (a-(c+1)) \\ =& \frac{c^7}{2240} - \frac{3c^5}{160} + \frac{c^4}{16} - \frac{3c^2}{10} - \frac{c}{2} + a - \frac{9}{35} \end{aligned} \tag{B.3.27}$$

Check: When $a = c+1$

$$\gamma_K(a,c) = \frac{85c^7-504c^6+798c^5+560c^4+840c^3-1680c^2+448c+1664}{2240} \tag{B.3.28}$$

ii. **Case 2** $c \in [-2,0]$**:** Under this condition, there are 4 cases that we need to consider:

a. Case 2(a): $a \leq -1$

$$\gamma_K(a,c) = 0 \tag{B.3.29}$$

b. Case 2(b): $a \in [-1, c+1]$

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{a} \frac{1}{16}(-t^3+3t+2)(-(t-c)^3+3t+2)\,dt \\ =& \frac{a\left(280c^3-840c+560\right)}{2240} + \frac{a^2\left(210c^3-420c^2-630c+840\right)}{2240} + \\ & \frac{105c^3+84c^2-280c+132}{2240} + \frac{a^4\left(-35c^3+420c-140\right)}{2240} + \\ & \frac{a^5\left(84c^2-168\right)}{2240} + \frac{a^3\left(-420c^2+280c+420\right)}{2240} - \frac{a^6c}{32} + \frac{a^7}{112} \end{aligned} \tag{B.3.30}$$

Check: When $a = c+1$

$$\gamma_K(a,c) = -\frac{c^7}{2240} + \frac{3c^5}{160} - \frac{c^3}{4} - \frac{3c^2}{10} + \frac{c}{2} + \frac{26}{35} \tag{B.3.31}$$

c. Case 2(c): $a \in [c+1, 1]$

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{1}{16}(-t^3+3t+2)(-(t-c)^3+3t+2)\,dt + \\ & \int_{c+1}^{a} \frac{-t^3+3t+2}{4}\,dt \\ =& -\frac{c^7}{2240} + \frac{3c^5}{160} + \frac{c^4}{16} - \frac{3c^2}{10} - \frac{c}{2} - \frac{a^4}{16} + \frac{3a^2}{8} + \frac{a}{2} - \frac{39}{560}.\end{aligned} \tag{B.3.32}$$

Check: When $a = 1$

$$\begin{aligned}\gamma_K(a,c) =& -\frac{c^7}{2240} + \frac{3c^5}{160} - \frac{c^3}{4} - \frac{3c^2}{10} + \frac{c}{2} + \frac{26}{35} + \frac{c^4}{16} + \frac{c^3}{4} - c \\ =& -\frac{c^7 - 42c^5 - 140c^4 + 672c^2 + 1120c - 1664}{2240}.\end{aligned} \tag{B.3.33}$$

d. Case 2(d) $a \geq 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{1}{16}(-t^3+3t+2)(-(t-c)^3+3t+2)\,dt + \\ & \int_{c+1}^{a} \frac{-t^3+3t+2}{4}\,dt + \int_{1}^{a} 1\,dt \\ =& \frac{-c^7 + 42c^5 + 140c^4 - 672c^2 - 1120c + 2240a - 576}{2240}.\end{aligned} \tag{B.3.34}$$

iii. **Case 3** $c \geq 2$**:** Under this condition, the intersection of the two CDF $\nu(t)$ and $\nu(t-c)$ can occur under the sub-cases below.

a. Case 3(a): $a \in [c-1, c+1]$

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{a} \frac{-(t-c)^3 + 3(t-c) + 2}{4}\,dt \\ =& -\frac{(-c+a-3)(-c+a+1)^3}{16}.\end{aligned} \tag{B.3.35}$$

b. Case 3(b): $a \geq c+1$

$$\gamma_K(a,c) = \int_{c-1}^{c+1} \frac{-(t-c)^3 + 3(t-c) + 2}{4}\,dt + \int_{c+1}^{a} 1\,dt = a - c. \tag{B.3.36}$$

iv. Case 4 $c \leq -2$: Under this condition, the overlapping of the two cdf $\nu(t)$ and $\nu(t-c)$ can occur when: When $c \leq 2$

a. Case 4(a): $a \in [-1, +1]$

$$\gamma_K(a, c) = \int_{c-1}^{a} \frac{-t^3 + 3t + 2}{4}\, dt = \frac{-a^4 + 6a^2 + 8a + 3}{16}. \tag{B.3.37}$$

b. Case 4(b): $a \geq c + 1$

$$\gamma_K(a, c) = \int_{c-1}^{c+1} \frac{-t^3 + 3t + 2}{4}\, dt + \int_{1}^{a} 1\, dt = a. \tag{B.3.38}$$

□

Derivation B.3.5. *Let K be Epanechnikov kernel as in Def B.3.1 with CDF ν_K of be as in B.3.1. The integration of two Epanechnikov complimentary CDF at different central points 0 and $c \in \mathbb{R}$ as in Lemma 5.4.2(ii) is*

i. For $c \in [0, 2]$

$$\xi_K(a, c) = \begin{cases} \frac{c^7-42c^5+140c^4-672c^2+1120c-2240a-576}{2240} & \textit{if } a \leq -1 \\ \frac{c^7-42c^5+140c^4-672c^2+1120(c-a)-140a^4+840a^2-156}{2240} & \textit{if } a \in [-1, c-1] \\ \frac{a\left(280c^3-840c-560\right)}{2240} + \frac{a^4\left(35c^3-420c-140\right)}{2240} + & \\ \frac{-105c^3+84c^2+280c+132}{2240} + \frac{a^2\left(-210c^3-420c^2+630c+840\right)}{2240} + & \\ \frac{a^3\left(420c^2+280c-420\right)}{2240} + \frac{a^5\left(168-84c^2\right)}{2240} + \frac{a^6 c}{32} - \frac{a^7}{112} & \textit{if } a \in [c-1, 1] \\ 0 & \textit{if } a \geq 1 \\ 0 & \textit{if } a \geq c+1 \end{cases}$$

ii. Case 1: $c \in [-2, 0]$

$$\xi_K(a, c) = \begin{cases} \frac{-c^7+42c^5+140c^4-672c^2+1120c-2240a-576}{2240} & \textit{if } a \leq c-1 \\ \frac{-c^7+42c^5+168c^2+560ac(c^2+a^2-3)-1120a+840a^2(1-c^2)-140a^4-156}{2240} & \textit{if } a \in [c-1, -1] \\ \frac{-c^7+42c^5-105c^3+84c^2+280c+132}{2240} + \frac{a\left(280c^3-840c-560\right)}{2240} + & \\ \frac{a^4\left(35c^3-420c-140\right)}{2240} + \frac{a^2\left(-210c^3-420c^2+630c+840\right)}{2240} + & \\ \frac{a^3\left(420c^2+280c-420\right)}{2240} + \frac{a^5\left(168-84c^2\right)}{2240} + \frac{a^6 c}{32} - \frac{a^7}{112} & \textit{if } a \in [-1, c+1] \\ 0 & \textit{if } a \geq c+1 \\ 0 & \textit{if } a \geq 1 \end{cases}$$

iii. Case 3: $c \geq 2$

$$\xi_K(a,c) = \begin{cases} -a & \textit{if} \quad a \leq -1 \\ \frac{-a^4+6a^2-8a+3}{16} & \textit{if} \quad a \in [-1,1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ 0 & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \geq c+1 \end{cases}$$

iv. Case 4: $c \leq -2$

$$\xi_K(a,c) = \begin{cases} c-a & \textit{if} \quad a \leq -1 \\ -\frac{(-c+a-1)^3(-c+a+3)}{16} & \textit{if} \quad a \in [-1,1] \\ 0 & \textit{if} \quad a \in [1, c-1] \\ 0 & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \geq c+1 \end{cases}$$

Proof. From Lemma 5.4.2, $\xi_K(a,c)$ is the reflection of $\gamma_K(a,c)$ on the y - axis. Hence, we can compute $\xi_K(a,c)$ by taking $\gamma_K(a,c)$. □

B.4 Quartic Kernel

Definition B.4.1. *Let the Quartic kernel be defined as*

$$K(u) = \begin{cases} \frac{15}{16}(1-u^2)^2 & \textit{if } |u| \leq 1 \\ 0 & \textit{otherwise.} \end{cases}$$

Derivation B.4.1. *Let K be the Quartic kernel function as in Def B.4.1. The CDF for Quartic kernel is*

$$\nu_K(t) = \begin{cases} 0 & \textit{if} \quad t \leq -1 \\ \frac{3t^5-10t^3+15t+8}{16} & \textit{if} \quad t \in [-1,1] \\ 1 & \textit{if} \quad t \geq 1 \end{cases}$$

Proof. Suppose K be a Quartic kernel as in Definition B.4.1. The cdf is the integration of the kernel function. There are three cases to be considered: (1) $t \leq -1$; (2) $t \in [-1,1]$; (3) $t \geq 1$.

1. Case 1 $a \leq -1$: Under this case, $\nu_K(t) = \int_{-\infty}^{-1} K(u)\, du = 0$ because the integration boundary is outside the limit where Quartic kernel is defined.
2. Case 2 $a \in [-1,1]$:

$$\nu_K(t) = \frac{15}{16}\int_{-1}^{a}(1-u^2)^2\, du = \frac{3t^5-10t^3+15t+8}{16} \tag{B.4.1}$$

3. Case 3 $a \geq 1$: Under this condition,

$$\nu_K(t) = 1 \tag{B.4.2}$$

□

Derivation B.4.2. *Let K be a Quartic kernel function as in Def B.4.1. The partial L2-product of two Quartic kernels at two different central points 0 and $c \in \mathbb{R}$ for the integration boundary is $(-\infty, a = \infty)$ is*

$$\lambda_K(c) = \begin{cases} \frac{5(-|c|^9+24c^7-336|c|^5+672c^4-768c^2+512)}{3584} & \textit{if} \quad |c| \leq 2 \\ 0 & \textit{if} \quad |c| > 2 \end{cases}$$

i.e.

$$\lambda_K(c) = \begin{cases} \frac{5(-|c|^9+24|c|^7-336|c|^5+672|c|^4-768|c|^2+512)}{3584} & \textit{if} \quad |c| \leq 2 \\ 0 & \textit{if} \quad |c| > 2 \end{cases}$$

Proof. Suppose K is a Quartic kernel as in Definition B.4.1. The partial L2-product of the Quartic kernel as in Eqn (5.4.2) is

$$\gamma_K(c) = \left(\frac{15}{16}\right)^2 \int (1-u^2)^2 \left(1-(u-c)^2\right)^2 \mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [-1,1])\, du$$

There are three cases to be considered: (1) $|c| \leq 2$; (2) $|c| \geq 2$. For (2), there is no intersection between the two kernels resulting $\gamma_K(a = \infty, c) = 0$. Hence, the computation focusses only (1).

1. **Case 1** $|c| \leq 2$: There are two cases that to be considered for this case.
   a. Case 1(a) $c \in [0,2]$:

$$\begin{aligned}\gamma_K(c) &= \left(\frac{15}{16}\right)^2 \int_{c-1}^{1} (1-u^2)\left(1-(u-c)^2\right)^2 du \\ &= \frac{5(-c^9+24c^7-336c^5+672c^4-768c^2+512)}{3584}\end{aligned} \tag{B.4.3}$$

   b. Case 1(b) $c \in [-2,0]$: When $-2 \leq c \leq 0$. The intersection happens between $[-1, c+1]$. Then, we can show that

$$\begin{aligned}\gamma_K(c) &= \left(\frac{15}{16}\right)^2 \int_{-1}^{c+1} (1-u^2)\left(1-(u-c)^2\right)^2 du \\ &= \frac{5(c^9-24c^7+336c^5+672c^4-768c^2+512)}{3584}\end{aligned} \tag{B.4.4}$$

□

Derivation B.4.3. *Let K be a Quartic kernel function as in Definition B.4.1. The partial L2-product of two Quartic kernels at two different central points 0 and $c \in \mathbb{R}$ is for the integration boundary is $(-\infty, a)$ where $a \in \mathbb{R}$ is*

1. *Case 1:* $c \in [0,2]$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq 0 \\ \frac{5(-c^9+24c^7-336c^5+336c^4+420c^3-384c^2-315c+256)}{3584} + \\ \frac{5(70a^9-315a^8c)}{3584} + \frac{5a^7(540c^2-360)}{3584} + \frac{5a^6(-420c^3+1260c)}{3584} + \\ \frac{5a^5(126c^4-1764c^2)}{3584} + \frac{5a^4(1260c^3-1890c+756)}{3584} + \\ \frac{5a^3(-420c^4+2100c^2-840)}{3584} + \frac{5a^2(-1260c^3+1260)}{3584} + \\ \frac{5a(630c^4-1260c^2+630)}{3584} & \textit{if } a \in [c-1,1] \\ \frac{5(-c^9+24c^7-336c^5+672c^4-768c^2+512)}{3584} & \textit{if } a \in [1,2) \end{cases}$$

2. *Case 2:* $c \in [-2,0]$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{25a^9}{256} - \frac{225a^8c}{512} + \frac{5a^7\left(540c^2-360\right)}{3584} + \frac{5a^6\left(1260c-420c^3\right)}{3584} + \\ \frac{5a^5\left(126c^4-1764c^2+756\right)}{3584} + \frac{5a^4\left(1260c^3-1890c\right)}{3584} + \\ \frac{5a^3\left(-420c^4+2100c^2-840\right)}{3584} + \frac{5a^2\left(1260c-1260c^3\right)}{3584} + \\ \frac{5a\left(630c^4-1260c^2+630\right)}{3584} + \frac{5\left(336c^4+420c^3-384c^2-315c+256\right)}{3584} & \textit{if } a \in [-1,c+1] \\ \frac{5(c^9-24c^7+336c^5+672c^4-768c^2+512)}{3584} & \textit{if } a \geq c+1 \end{cases}$$

Proof. Suppose K is a Quartic kernel as in Definition B.4.1. The partial L2-product of the Quartic kernel as in Eqn (5.4.2) is

$$\lambda_K(a,c) = \left(\frac{15}{16}\right)^2 \int_{\infty}^{a} (1-u^2)^2 \left(1-(u-c)^2\right)^2 \mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [-1,1])\, du \tag{B.4.5}$$

There are three cases to be considered: (1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $|c| \geq 2$. For (3), this is outside range of intersection of the two Quartic kernels resulting $\lambda_K(a,c) = 0$. Hence, the computation of $\lambda_K(a,c)$ for Quartic kernel only focusses of case (1) and (2).

1. **Case 1** $c \in [0,2]$**:** The intersection between the two kernels happens between $[c-1,1]$. Under this condition, we need to consider the cases below:

 a. Case 1(a) $a \leq c-1$:

$$\lambda_K(a,c) = 0 \tag{B.4.6}$$

b. Case 1(b) $a \in [c-1, 1]$:

$$\begin{aligned}\lambda_K(a,c) &= \left(\frac{15}{16}\right)^2 \int_{c-1}^{a} (1-u^2)^2(1-(u-c)^2)^2\,du\\ &=\frac{5(-c^9+24c^7-336c^5+336c^4+420c^3-384c^2-315c+256)}{3584}+\\ &\frac{5(70a^9-315a^8c)}{3584}+\frac{5a^7(540c^2-360)}{3584}+\frac{5a^6(-420c^3+1260c)}{3584}+\\ &\frac{5a^5(126c^4-1764c^2)}{3584}+\frac{5a^4(1260c^3-1890c+756)}{3584}+\\ &\frac{5a^3(-420c^4+2100c^2-840)}{3584}+\frac{5a^2(-1260c^3+1260)}{3584}+\\ &\frac{5a(630c^4-1260c^2+630)}{3584}\end{aligned} \tag{B.4.7}$$

c. Case 1(c) $a \geq 1$:

$$\begin{aligned}\lambda_K(a,c) &= \left(\frac{15}{16}\right)^2 \int_{c-1}^{1} (1-u^2)^2(1-(u-c)^2)^2\,du\\ &=\frac{5(-c^9+24c^7-336c^5+672c^4-768c^2+512)}{3584}\end{aligned} \tag{B.4.8}$$

2. **Case 2** $c \in [-2, 0]$**:**The intersection between the two kernels happens between $[-1, c+1]$. Under this condition, we need to consider the cases below.

a. Case 2(a) $a < -1$:

$$\lambda_K(a,c) = 0 \tag{B.4.9}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}\lambda_K(a,c) &= \int_{-1}^{a} \left(\frac{15}{16}\right)^2 (1-u^2)^2(1-(u-c)^2)^2\,du\\ &=\frac{25a^9}{256}-\frac{225a^8c}{512}+\frac{5a^7\left(540c^2-360\right)}{3584}+\frac{5a^6\left(1260c-420c^3\right)}{3584}+\\ &\frac{5a^5\left(126c^4-1764c^2+756\right)}{3584}+\frac{5a^4\left(1260c^3-1890c\right)}{3584}+\\ &\frac{5a^3\left(-420c^4+2100c^2-840\right)}{3584}+\frac{5a^2\left(1260c-1260c^3\right)}{3584}+\\ &\frac{5a\left(630c^4-1260c^2+630\right)}{3584}+\frac{5\left(336c^4+420c^3-384c^2-315c+256\right)}{3584}\end{aligned} \tag{B.4.10}$$

c. Case 2(c) $a \geq c+1$:

$$\lambda_K(a,c) = \left(\frac{15}{16}\right)^2 \int_{-1}^{c+1} (1-u^2)\left(1-(u-c)^2\right)^2\, du$$
$$= \frac{5(c^9 - 24c^7 + 336c^5 + 672c^4 - 768c^2 + 512)}{3584} \tag{B.4.11}$$

□

Derivation B.4.4. *Let K be a Quartic kernel function as in Def B.4.1, then the partial L2-product of the CDF at two different central points 0 and $c \in \mathbb{R}$ for the integral boundary $(-\infty, a)$ where $a \in \mathbb{R}$ is*

1. For $c \in [0,2]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if}\ a \leq c-1 \\ \frac{3c^{11}-110c^9+2640c^7+\left(-1386a^6+6930a^4-20790a^2-22176a-6930\right)c^5}{236544}+ & \\ \frac{\left(5940a^7-27720a^5+69300a^3+55440a^2-7920\right)c^4}{236544}+ & \\ \frac{\left(-10395a^8+50820a^6-127050a^4-73920a^3+69300a^2+73920a+17325\right)c^3}{236544}+ & \\ \frac{\left(9240a^9-51480a^7+138600a^5+55440a^4-138600a^3-110880a^2+13200\right)c^2}{236544}+ & \\ \frac{\left(-4158a^{10}+27720a^8-87780a^6-22176a^5+138600a^4+\right)c}{236544}+ & \\ \frac{(73920a^3-103950a^2-110880a-29568)c}{236544}+ & \\ \frac{756a^{11}-6160a^9+25080a^7+7392a^6-55440a^5-36960a^4+}{236544} & \\ \frac{69300a^3+110880a^2+59136a+11360}{236544} & \textit{if}\ a \in [c-1,1] \\ \frac{3c^{11}-110c^9+2640c^7-44352ac^5+\left(110880a^2-15840\right)c^4+}{236544}+ & \\ \frac{\left(-147840a^3+147840a\right)c^3}{236544}+ & \\ \frac{\left(110880a^4-221760a^2+26400\right)c^2+\left(-44352a^5+147840a^3-221760a\right)c}{236544}+ & \\ \frac{7392a^6-36960a^4+110880a^2+118272a-14240}{236544} & \textit{if}\ a \in [1,c+1] \\ \frac{3c^{11}-110c^9+2640c^7-7392c^6+44352c^5-89760c^4+26400c^2-250784+236544a}{236544} & \textit{if}\ a \geq c+1 \end{cases}$$

2. *For* $c \in [-2, 0]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ -\frac{\left(1386a^6-6930a^4+20790a^2+22176a\right)c^5}{236544} - \\ \frac{\left(-5940a^7+27720a^5-69300a^3-55440a^2\right)c^4}{236544} - \\ \frac{\left(10395a^8-50820a^6+127050a^4+73920a^3-69300a^2-73920a\right)c^3}{236544} - \\ \frac{\left(-9240a^9+51480a^7-138600a^5-55440a^4+138600a^3+110880a^2\right)c^2}{236544} - \\ \frac{\left(4158a^{10}-27720a^8+87780a^6+22176a^5-138600a^4-73920a^3\right)c^2}{236544} + \\ \frac{\left(+103950a^2+110880a\right)c}{236544} - \\ \frac{(-756a^{11}+6160a^9-25080a^7-7392a^6+55440a^5+36960a^4-69300a^3)}{236544} + \\ \frac{-110880a^2-59136a+6930c^5-7920c^4+17325c^3+13200c^2-29568c+11360)}{236544} & \textit{if } a \in [-1, c+1] \\ \frac{-3c^{11}+110c^9-2640c^7-7392c^6+21120c^4-84480c^2-118272c}{236544} + \\ \frac{7392a^6-36960a^4+110880a^2+118272a-14240}{236544} & \textit{if } a \in [c+1, 1] \\ \frac{-3c^{11}+110c^9-2640c^7-14784c^6-44352c^5-52800c^4-84480c^2}{236544} + \\ \frac{-354816c+236544a-236544}{236544} & \textit{if } a \geq 1 \end{cases}$$

3. *For* $c \geq 2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{(a-c+1)^4\left(a^2-2a(c+2)+c^2+4c+5\right)}{32} & \textit{if } a \in [c-1, c+1] \\ a-c & \textit{if } a \geq c+1 \end{cases}$$

4. *For* $c \leq -2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{a^6}{32} - \frac{5a^4}{32} + \frac{15a^2}{32} + \frac{a}{2} + \frac{5}{32} & \textit{if } a \in [-1, 1] \\ a & \textit{if } a \geq 1 \end{cases}$$

Proof. Let K be a Quartic kernel as in Def B.4.1. The CDF is defined in B.4.1. Then, the partial L2-product of CDF as in Eqn (5.4.3) is

$$\gamma_K(a,c) = \int_{-\infty}^{t} (1-u^2)^2(1-(u-c)^2)^2 \mathbb{1}(u \in [-1,1]) \mathbb{1}(u \in [c-1, c+1]) \, du$$

1. **Case 1** $c \in [0, 2]$**:** Under this condition, we need to consider 4 sub-cases below.

a. Case 1(a) $a \leq c-1$:

$$\gamma_K(a,c) = \int_{-\infty}^{a} \nu_K(t)\nu_K(t-c)\,dt = 0 \tag{B.4.12}$$

b. Case 1(b) $a \in [c-1, 1]$:

$$\begin{aligned}
&\gamma_K(a,c) \\
=&\int_{c-1}^{a} \left(\frac{3t^5-10t^3+15t+8}{16}\right)\left(\frac{3(t-c)^5-10(t-c)^3+15(t-c)+8}{16}\right)\,dt \\
=&\frac{3c^{11}-110c^9+2640c^7}{236544}+\frac{\left(-1386a^6+6930a^4-20790a^2-22176a-6930\right)c^5}{236544} \\
&\frac{\left(5940a^7-27720a^5+69300a^3+55440a^2-7920\right)c^4}{236544}+ \\
&\frac{\left(-10395a^8+50820a^6-127050a^4-73920a^3+69300a^2+73920a+17325\right)c^3}{236544}+ \\
&\frac{\left(9240a^9-51480a^7+138600a^5+55440a^4-138600a^3-110880a^2+13200\right)c^2}{236544}+ \\
&\frac{(-4158a^{10}+27720a^8-87780a^6-22176a^5+138600a^4)c}{236544} \\
&\frac{(73920a^3-103950a^2-110880a-29568)c}{236544}+ \\
&\frac{756a^{11}-6160a^9+25080a^7+7392a^6-55440a^5-36960a^4+69300a^3}{236544}+ \\
&\frac{110880a^2+59136a+11360}{236544}.
\end{aligned} \tag{B.4.13}$$

Check:When $a=1$,

$$=\frac{3c^{11}-110c^9+2640c^7-44352c^5+95040c^4-84480c^2-118272c+185344}{236544} \tag{B.4.14}$$

c. Case 1(c) $a \in [1, c+1]$:

$$\begin{aligned}
&\gamma_K(a,c) \\
=&\int_{c-1}^{1} \left(\frac{3t^5-10t^3+15t+8}{16}\right)\left(\frac{3(t-c)^5-10(t-c)^3+15(t-c)+8}{16}\right)\,dt+ \\
&\int_{1}^{a} \frac{3(t-c)^5-10(t-c)^3+15(t-c)+8}{16}\,dt \\
=&\frac{3c^{11}-110c^9+2640c^7-44352c^5+95040c^4-84480c^2-118272c+185344}{236544}+ \\
&\frac{-30ac+30c-(c-1)^6+5\,(c-1)^4+(a-c)^6-5\,(a-c)^4+15a^2+16a-31}{32}.
\end{aligned} \tag{B.4.15}$$

Check: When $a = c + 1$

$$\begin{aligned}\gamma_K(a,c) =& \frac{3c^{11} - 110c^9 + 2640c^7 - 7392c^6 + 44352c^5 - 89760c^4 + 26400c^2}{236544} + \\ & \frac{236544c - 14240}{236544}\end{aligned} \tag{B.4.16}$$

d. Case 1(d) $a \geq c + 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{1} \left(\frac{3t^5 - 10t^3 + 15t + 8}{16}\right)\left(\frac{3(t-c)^5 - 10(t-c)^3 + 15(t-c) + 8}{16}\right) dt + \\ & \int_{1}^{c+1} \frac{3(t-c)^5 - 10(t-c)^3 + 15(t-c) + 8}{16}\, dt + \int_{c+1}^{a} 1\, dt \\ =& \frac{3c^{11} - 110c^9 + 2640c^7 - 7392c^6 + 44352c^5 - 89760c^4}{236544} + \\ & \frac{26400c^2 - 250784 + 236544a}{236544}\end{aligned} \tag{B.4.17}$$

2. **Case 2** $c \in [-2, 0]$**:** Under this condition, we consider 4 sub-cases below.

a. Case 2(a) $a \leq -1$:

$$\gamma_K(a,c) = \int_{-\infty}^{a} \nu(t)\nu(t-c)\, dt = 0 \tag{B.4.18}$$

b. Case 2(b) $a \in [-1, c + 1]$:

$$\begin{aligned}&\gamma_K(a,c) \\ =& \int_{-1}^{a} \left(\frac{3t^5 - 10t^3 + 15t + 8}{16}\right)\left(\frac{3(t-c)^5 - 10(t-c)^3 + 15(t-c) + 8}{16}\right) dt \\ =& -\frac{\left(1386a^6 - 6930a^4 + 20790a^2 + 22176a\right)c^5}{236544} - \\ & \frac{\left(-5940a^7 + 27720a^5 - 69300a^3 - 55440a^2\right)c^4}{236544} - \\ & \frac{\left(10395a^8 - 50820a^6 + 127050a^4 + 73920a^3 - 69300a^2 - 73920a\right)c^3}{236544} - \\ & \frac{\left(-9240a^9 + 51480a^7 - 138600a^5 - 55440a^4 + 138600a^3 + 110880a^2\right)c^2}{236544} - \\ & \frac{\left(4158a^{10} - 27720a^8 + 87780a^6 + 22176a^5 - 138600a^4 - 73920a^3\right)c^2}{236544} + \\ & \frac{\left(+103950a^2 + 110880a\right)c}{236544} - \\ & \frac{(-756a^{11} + 6160a^9 - 25080a^7 - 7392a^6 + 55440a^5 + 36960a^4 - 69300a^3)}{236544} + \\ & \frac{-110880a^2 - 59136a + 6930c^5 - 7920c^4 + 17325c^3 + 13200c^2 - 29568c + 11360)}{236544}\end{aligned} \tag{B.4.19}$$

Check: When $a = c + 1$

$$\begin{aligned}&\gamma_K(a,c)\\=&\frac{-3c^{11}+110c^9-2640c^7+44352c^5+95040c^4-84480c^2}{236544}+\\&\frac{118272c+185344}{236544}\end{aligned}\tag{B.4.20}$$

c. Case 2(c) $a \in [c + 1, 1]$:

$$\begin{aligned}&\gamma_K(a,c)\\=&\int_{-1}^{c+1}\left(\frac{3t^5-10t^3+15t+8}{16}\right)\left(\frac{3(t-c)^5-10(t-c)^3+15(t-c)+8}{16}\right)dt+\\&\int_{c+1}^{a}\frac{3t^5-10t^3+15t+8}{16}\,dt\\=&\frac{-3c^{11}+110c^9-2640c^7-7392c^6+21120c^4-84480c^2-118272c}{236544}+\\&\frac{7392a^6-36960a^4+110880a^2+118272a-14240}{236544}\end{aligned}\tag{B.4.21}$$

Check: When $a = 1$

$$\begin{aligned}&\gamma_K(a,c)\\=&\frac{(-3c^{11}+110c^9-2640c^7-14784c^6-44352c^5-52800c^4-84480c^2-354816c)}{236544}\end{aligned}\tag{B.4.22}$$

d. Case 2(d) $a \geq 1$:

$$\begin{aligned}&\gamma_K(a,c)\\=&\int_{-1}^{c+1}\left(\frac{3t^5-10t^3+15t+8}{16}\right)\left(\frac{3(t-c)^5-10(t-c)^3+15(t-c)+8}{16}\right)dt+\\&\int_{c+1}^{1}\frac{3t^5-10t^3+15t+8}{16}\,dt+\int_{1}^{a}1\,dt\\=&\frac{-3c^{11}+110c^9-2640c^7-14784c^6-44352c^5-52800c^4-84480c^2}{236544}+\\&\frac{-354816c+236544a-236544}{236544}\end{aligned}\tag{B.4.23}$$

3. **Case 3:** $c \geq 2$**:** Under this condition, we consider the sub-cases below.

a. Case 3(a) $a \in [c-1, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{a} \frac{3(t-c)^5 - 10(t-c)^3 + 15(t-c) + 8}{16}\, dt \\ =& \frac{(a-c+1)^4\,(a^2 - 2a(c+2) + c^2 + 4c + 5)}{32} \end{aligned} \tag{B.4.24}$$

b. Case 3(b) $a \geq 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{c+1} \left(\frac{3(t-c)^5 - 10(t-c)^3 + 15(t-c) + 8}{16}\right) dt + \int_{c+1}^{a} 1\, dt \\ =& a - c \end{aligned} \tag{B.4.25}$$

4. **Case 4:** $c \geq 2$**:** Under this condition, we consider the sub-cases below.

a. Case 4(a) $a \in [-1, 1]$:

$$\gamma_K(a,c) = \int_{-1}^{a} \left(\frac{3t^5 - 10t^3 + 15t + 8}{16}\right) dt = \frac{a^6}{32} - \frac{5a^4}{32} + \frac{15a^2}{32} + \frac{a}{2} + \frac{5}{32} \tag{B.4.26}$$

b. Case 4(b) $a \geq 1$:

$$\gamma_K(a,c) = \int_{-1}^{1} \left(\frac{3t^5 - 10t^3 + 15t + 8}{16}\right) dt + \int_{1}^{a} 1\, dt = a \tag{B.4.27}$$

□

Derivation B.4.5. *Let K be a Quartic kernel and let the CDF and partial L2-product of the CDF as in B.4.1. The partial L2-product of $1 - \nu_K(t)$ for K at two different central points, 0 and $c \in \mathbb{R}$ for the integral boundary of $[a, \infty)$ where $a \in \mathbb{R}$ is*

1. For $c \in [0, 2]$:

$$\xi_K(a,c) = \int_{a}^{\infty} (1 - \nu_K(t))(1 - \nu_K(t-c))\, dt$$

$$
= \begin{cases}
\frac{3c^{11}-110c^9+2640c^7-7392c^6+21120c^4-84480c^2+118272c}{236544}+ & \\
\frac{7392a^6-36960a^4+110880a^2-354816a-250784}{236544} & \textit{if } a \leq -1 \\
\frac{3c^{11}-110c^9+2640c^7-7392c^6+21120c^4-84480c^2+118272c}{236544}+ & \\
\frac{7392a^6-36960a^4+110880a^2-118272a-14240}{236544} & \textit{if } a \in [-1, c-1] \\
\frac{\left(1386a^6-6930a^4+20790a^2-22176a\right)c^5}{236544}+ & \\
\frac{\left(-5940a^7+27720a^5-69300a^3+55440a^2\right)c^4}{236544}+ & \\
\frac{\left(10395a^8-50820a^6+127050a^4-73920a^3-69300a^2+73920a\right)c^3}{236544}- & \\
\frac{\left(9240a^9-51480a^7+138600a^5-55440a^4-138600a^3+110880a^2\right)c^2}{236544}+ & \\
\frac{\left(4158a^{10}-27720a^8+87780a^6-22176a^5-138600a^4+73920a^3\right)c}{236544}+ & \\
\frac{\left(103950a^2-110880a\right)c}{236544}- & \\
\frac{\substack{(756a^{11}-6160a^9+25080a^7-7392a^6-55440a^5+36960a^4+69300a^3) \\ + \\ (-110880a^2+59136a+6930c^5-7920c^4-17325c^3+13200c^2+29568c+11360)}}{236544} & \textit{if } a \in [c-1, 1] \\
0 & \textit{if } a \geq 1
\end{cases}
$$

2. *For* $c \in [-2, 0]$

$$
= \begin{cases}
\frac{-3c^{11}+110c^9-2640c^7-7392c^6-44352c^5-89760c^4+26400c^2}{236544}+ & \\
\frac{-250784-236544a}{236544} & \textit{if } a \leq c-1 \\
\frac{-3c^{11}+110c^9-2640c^7-44352ac^5}{236544} + \frac{\left(110880a^2-15840\right)c^4}{236544}+ & \\
\frac{\left(147840a-147840a^3\right)c^3}{236544} \frac{\left(110880a^4-221760a^2+26400\right)c^2}{236544}+ & \\
\frac{\left(-44352a^5+147840a^3-221760a\right)c}{236554}+ & \\
\frac{7392a^6-36960a^4+110880a^2-118272a-14240}{236544} & \textit{if } a \in [c-1, -1] \\
\frac{-3c^{11}+110c^9-2640c^7}{236544} + \frac{\left(1386a^6-6930a^4+20790a^2-22176a+6930\right)c^5}{236544}+ & \\
\frac{\left(-5940a^7+27720a^5-69300a^3+55440a^2-7920\right)c^4}{236544}+ & \\
\frac{\left(10395a^8-50820a^6+127050a^4-73920a^3-69300a^2+73920a-17325\right)c^3}{236544}+ & \\
\frac{\left(-9240a^9+51480a^7-138600a^5+55440a^4+138600a^3-110880a^2+13200\right)c^2}{236544}+ & \\
\frac{\left(4158a^{10}-27720a^8+87780a^6-22176a^5-138600a^4+73920a^3+103950a^2\right)c}{236544}+ & \\
\frac{(-110880a+29568)c}{236544}+ & \\
\frac{-756a^{11}+6160a^9-25080a^7+7392a^6+55440a^5-36960a^4-69300a^3}{236544}+ & \\
\frac{110880a^2-59136a+11360}{236544} & \textit{if } a \in [-1, c+1] \\
0 & \textit{if } a \geq c+1
\end{cases}
$$

3. *For* $c \geq 2$

$$= \begin{cases} -a & \textit{if} \quad a \leq -1 \\ -\frac{2a^6+5a^4-32a-39}{64} & \textit{if} \quad a \in [-1, 1] \\ 0 & \textit{if} a \geq 1 \end{cases}$$

4. *For* $c \leq -2$

$$= \begin{cases} c - a & \textit{if} \quad a \leq c - 1 \\ -\frac{(c-a-1)\big((c-a)\big((c-a)(c-a+1)\big(2(c-a)^2+7\big)+7\big)+39\big)}{64} & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \geq 1 \end{cases}$$

Proof. The derivation can be obtained by reflection of the partial L2-product of CDF. For $\xi(a, c > 0)$, it can be found by looking at $\gamma(a, c \leq 0)$ and change the signs for a and c. For $\xi(a, c \leq 0)$, it can be found by looking at $\gamma(a, c \geq 0)$ and change the signs for a and c. □

B.5 Triweight Kernel

Definition B.5.1. *Let the Triweight kernel be defined as*

$$K(u) = \begin{cases} \frac{35}{32}(1-u^2)^3 & \textit{if} \quad |u| \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

Derivation B.5.1. *Let K be the Triweight kernel as in Def B.5.1. The CDF of the Triweight kernel is*

$$\nu_K(t) = \begin{cases} 0 & \textit{if} \quad t \leq -1 \\ \frac{-5t^7+21t^5-35t^3+35t+16}{32} & \textit{if} \quad t \in [-1,1] \\ 1 & \textit{if} \quad t \geq 1 \end{cases}$$

Proof. Suppose K is a Triweight kernel as in Def B.5.1. The CDF is the integration of the kernel function. There are three cases to considered: (1) $t \leq -1$ (2) $t \in [-1,1]$ (3) $t\, ge 1$.

i. **Case 1** $t \leq -1$**:** Under this case, the integration of K is outside the range of Definition B.5.1 resulting $\nu_K(t) = 0$.

ii. **Case 2** $t \in [-1,1]$**:**

$$\nu_K(t) = \frac{35}{32}\int_{-1}^{t}(1-u^2)^3\, du = \frac{-5t^7+21t^5-35t^3+35t+16}{32} \tag{B.5.1}$$

iii. **Case 3** $t \geq 1$**:**

$$\nu_K(t) = \frac{35}{32}\int_{-1}^{1}(1-u^2)^3\, du = 1 \tag{B.5.2}$$

□

Derivation B.5.2. *Let K be a Triweight kernel as in Def B.5.1. The partial L2-product of two Triweight kernels at two different central points 0 and $c \in \mathbb{R}$ is for the integration boundary is $(-\infty, \infty)$ is*

$$\lambda_K(c) = \begin{cases} \frac{-175|c|^{13}+5460|c|^{11}-80080|c|^9+960960|c|^7-1921920|c|^6+}{1757184} & \\ \frac{2562560|c|^4-2795520|c|^2+1433600}{1757184} & \textit{if} \quad |c| \leq 2 \\ 0 & \textit{if} \quad |c| > 2 \end{cases}$$

Proof. Suppose K is a Triweight kernel as in Def B.5.1. The partial L2-product of the Triweight kernel as in Eqn (5.4.2) is

$$\begin{aligned}\lambda_K(c) &= \left(\frac{35}{32}\right)^2 \int_{-\infty}^{\infty} (1-u^2)^3 \left(1-u^2\right)^3 \mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [-1,1])\, du \\ &= \left(\frac{35}{32}\right)^2 \int_{-\infty}^{a} (1-u^2)^3 \left(1-u^2\right)^3 \, du \qquad \text{(B.5.3)}\end{aligned}$$

There are three cases to be considered: (1) $|c| \leq 2$; (2) $|c| > 2$. For (2), there is no intersection between the two kernels resulting $\gamma_K(a = \infty, c) = 0$. Hence, the computation focusses only (1).

i. Case 1(a) $c \in [c-1, 1$:

$$\begin{aligned}\lambda_K(c) &= \int_{c-1}^{1} \left(\frac{35}{32}\right)^2 \int_{-1}^{1} (1-u^2)^3 \left(1-u^2\right)^3 \, du \\ &= \frac{-175c^{13} + 5460c^{11} - 80080c^9 + 960960c^7 - 1921920c^6 +}{1757184} \\ &\quad \frac{2562560c^4 - 2795520c^2 + 1433600}{1757184} \qquad \text{(B.5.4)}\end{aligned}$$

ii. Case 1(b) $c \in [-1, c-1]$:

$$\begin{aligned}\lambda_K(c) &= \int_{-1}^{c+1} \left(\frac{35}{32}\right)^2 \int_{-1}^{1} (1-u^2)^3 \left(1-u^2\right)^3 \, du \\ &= \frac{175c^{13} - 5460c^{11} + 80080c^9 - 960960c^7 - 1921920c^6 +}{1757184} \\ &\quad \frac{2562560c^4 - 2795520c^2 + 1433600}{1757184} \qquad \text{(B.5.5)}\end{aligned}$$

□

Derivation B.5.3. *Let K be a Triweight kernel as in Def B.5.1. The partial L2-product of two Triweight kernels at two different central points 0 and $c \in \mathbb{R}$is $(-\infty, a)$ where $a \in \mathbb{R}$, $\lambda_K(a, c)$ is*

i. ***Case 1*** $c \in [0, 2]$***:***

$$= \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{1225a^{13}}{13312} - \frac{1225a^{12}c}{2048} + \frac{35a^{11}\left(81900c^2-32760\right)}{1757184} + \\ \frac{35a^{10}\left(180180c-120120c^3\right)}{1757184} + \frac{35a^9\left(100100c^4-420420c^2+100100\right)}{1757184} + \\ \frac{35a^8\left(-45045c^5+540540c^3-450450c\right)}{1757184} + \frac{35a^7\left(8580c^6-411840c^4+875160c^2-171600\right)}{1757184} + \\ \frac{35a^6\left(180180c^5-960960c^3+600600c\right)}{1757184} + \frac{35a^5\left(-36036c^6+648648c^4-936936c^2+180180\right)}{1757184} + \\ \frac{35a^4\left(-270270c^5+840840c^3-450450c\right)}{1757184} + \\ \frac{35a^3\left(60060c^6-480480c^4+540540c^2-120120\right)}{1757184} + \frac{35a^2\left(180180c^5-360360c^3+180180c\right)}{1757184} + \\ \frac{35a\left(-60060c^6+180180c^4-180180c^2+60060\right)}{1757184} + \\ \frac{35\left(-5c^{13}+156c^{11}-2288c^9+27456c^7-27456c^6-45045c^5+36608c^4\right)}{1757184} + \\ \frac{35\left(60060c^3-39936c^2-30030c+20480\right)}{1757184} & \textit{if } a \in [c-1, 1] \\ \frac{-175c^{13}+5460c^{11}-80080c^9+960960c^7-1921920c^6}{1757184} + \\ \frac{2562560c^4-2795520c^2+1433600}{1757184} & \textit{if } a \geq 1 \end{cases}$$

ii. ***Case 2*** $c \in [-2, 0]$***:***

$$= \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{1225a^{13}}{13312} - \frac{1225a^{12}c}{2048} + \frac{35a^{11}\left(81900c^2-32760\right)}{1757184} + \\ \frac{35a^{10}\left(180180c-120120c^3\right)}{1757184} + \frac{35a^9\left(100100c^4-420420c^2+100100\right)}{1757184} + \\ \frac{35a^8\left(-45045c^5+540540c^3-450450c\right)}{1757184} + \frac{35a^7\left(8580c^6-411840c^4+875160c^2-171600\right)}{1757184} + \\ \frac{35a^6\left(180180c^5-960960c^3+600600c\right)}{1757184} + \frac{35a^5\left(-36036c^6+648648c^4-936936c^2+180180\right)}{1757184} + \\ \frac{35a^4\left(-270270c^5+840840c^3-450450c\right)}{1757184} + \frac{35a^3\left(60060c^6-480480c^4+540540c^2-120120\right)}{1757184} + \\ \frac{35a^2\left(180180c^5-360360c^3+180180c\right)}{1757184} + \\ \frac{35a\left(-60060c^6+180180c^4-180180c^2+60060\right)}{1757184} + \\ \frac{35\left(-27456c^6-45045c^5+36608c^4+60060c^3-39936c^2-30030c+20480\right)}{1757184} & \textit{if } a \in [-1, c+1] \\ \frac{175c^{13}-5460c^{11}+80080c^9-960960c^7-1921920c^6}{1757184} + \\ \frac{2562560c^4-2795520c^2+1433600}{1757184} & \textit{if } a \geq c+1 \end{cases}$$

Proof. Suppose K is a Triweight kernel as in Def B.5.1. The partial L2-product of the Triweight kernel as in Eqn (5.4.2) is

$$\begin{aligned}\lambda_K(a,c) &= \left(\frac{35}{32}\right)^2 \int_{-\infty}^{a} (1-u^2)^3 \left(1-(u-c)^2\right)^3 \mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [-1,1])\, du \\ &= \left(\frac{35}{32}\right)^2 \int_{-\infty}^{a} (1-(u-c)^2)^3 \left(1-u^2\right)^3\; du \qquad \text{(B.5.6)}\end{aligned}$$

There are three cases to be considered: (1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $|c| \geq 2$. For (3), this is outside the intersection of the two Triweight kernels resulting $\lambda_K(a,c) = 0$. Hence, the computation of $\lambda_K(a,c)$ for Triweight kernel only focusses on the two case (1) and (2).

i. **Case 1** $c \in [0,2]$**:** The intersection between the two kernels occurs in $[c-1,1]$. Under the condition, we need to consider the cases below:

a. Cases 1(a) $a \leq c-1$:

$$\lambda_K(a,c) = 0 \qquad \text{(B.5.7)}$$

b. Case 1(b) $a \in [c-1, 1]$:

$$
\begin{aligned}
\lambda_K(a,c) =& \int_{c-1}^{a} \left(\frac{35}{32}\right)^2 (1-u^2)^3(1-(u-c)^2)^3\, du \\
=&\frac{1225a^{13}}{13312} - \frac{1225a^{12}c}{2048} + \frac{35a^{11}\left(81900c^2 - 32760\right)}{1757184} + \\
&\frac{35a^{10}\left(180180c - 120120c^3\right)}{1757184} + \frac{35a^{9}\left(100100c^4 - 420420c^2 + 100100\right)}{1757184} + \\
&\frac{35a^{8}\left(-45045c^5 + 540540c^3 - 450450c\right)}{1757184} + \\
&\frac{35a^{7}\left(8580c^6 - 411840c^4 + 875160c^2 - 171600\right)}{1757184} + \\
&\frac{35a^{6}\left(180180c^5 - 960960c^3 + 600600c\right)}{1757184} + \\
&\frac{35a^{5}\left(-36036c^6 + 648648c^4 - 936936c^2 + 180180\right)}{1757184} + \\
&\frac{35a^{4}\left(-270270c^5 + 840840c^3 - 450450c\right)}{1757184} + \\
&\frac{35a^{3}\left(60060c^6 - 480480c^4 + 540540c^2 - 120120\right)}{1757184} + \\
&\frac{35a^{2}\left(180180c^5 - 360360c^3 + 180180c\right)}{1757184} + \\
&\frac{35a\left(-60060c^6 + 180180c^4 - 180180c^2 + 60060\right)}{1757184} + \\
&\frac{35\left(-5c^{13} + 156c^{11} - 2288c^9 + 27456c^7 - 27456c^6 - 45045c^5 + 36608c^4\right) +}{1757184} \\
&\frac{35\left(60060c^3 - 39936c^2 - 30030c + 20480\right)}{1757184} \qquad \text{(B.5.8)}
\end{aligned}
$$

c. Case 1(c) $a \geq 1$:

$$
\begin{aligned}
\lambda_K(a,c) =& \int_{c-1}^{1} \left(\frac{35}{32}\right)^2 (1-u^2)^3(1-(u-c)^2)^3\, du \\
=&\frac{-175c^{13} + 5460c^{11} - 80080c^9 + 960960c^7 - 1921920c^6 +}{1757184} \\
&\frac{2562560c^4 - 2795520c^2 + 1433600}{1757184} \qquad \text{(B.5.9)}
\end{aligned}
$$

ii. **Case 2** $c \in [-2, 0]$**:**The intersection between the two kernels happens between $[-1, c+1]$. Under this condition, we need to consider the cases below:

a. Case 2(a) $a \leq -1$

$$\lambda_K(a,c) = 0 \qquad \text{(B.5.10)}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}
&\lambda_K(a,c)\\
=&\int_{-1}^{a} \left(\frac{35}{32}\right)^2 (1-u^2)^3(1-(u-c)^2)^3\,du\\
=&\frac{1225a^{13}}{13312} - \frac{1225a^{12}c}{2048} + \frac{35a^{11}\left(81900c^2 - 32760\right)}{1757184} +\\
&\frac{35a^{10}\left(180180c - 120120c^3\right)}{1757184} + \frac{35a^9\left(100100c^4 - 420420c^2 + 100100\right)}{1757184} +\\
&\frac{35a^8\left(-45045c^5 + 540540c^3 - 450450c\right)}{1757184} +\\
&\frac{35a^7\left(8580c^6 - 411840c^4 + 875160c^2 - 171600\right)}{1757184} +\\
&\frac{35a^6\left(180180c^5 - 960960c^3 + 600600c\right)}{1757184} +\\
&\frac{35a^5\left(-36036c^6 + 648648c^4 - 936936c^2 + 180180\right)}{1757184} +\\
&\frac{35a^4\left(-270270c^5 + 840840c^3 - 450450c\right)}{1757184} +\\
&\frac{35a^3\left(60060c^6 - 480480c^4 + 540540c^2 - 120120\right)}{1757184} +\\
&\frac{35a^2\left(180180c^5 - 360360c^3 + 180180c\right)}{1757184} +\\
&\frac{35a\left(-60060c^6 + 180180c^4 - 180180c^2 + 60060\right)}{1757184} +\\
&\frac{35\left(-27456c^6 - 45045c^5 + 36608c^4 + 60060c^3 - 39936c^2 - 30030c + 20480\right)}{1757184}
\end{aligned} \tag{B.5.11}$$

c. Case 2(c) $a \geq 1$:

$$\begin{aligned}
\lambda_K(a,c) =&\frac{175c^{13} - 5460c^{11} + 80080c^9 - 960960c^7 - 1921920c^6 +}{1757184}\\
&\frac{2562560c^4 - 2795520c^2 + 1433600}{1757184}
\end{aligned} \tag{B.5.12}$$

□

Proposition B.5.1. *Let K be the Triweight kernel as in Def B.5.1. The integral of two Triweight CDF with two different central points a and c from $-\infty$ to $a \in \mathbb{R}$ is*

$$\gamma_K(a,c) \tag{B.5.13}$$

1. *For* $c \in [0,2]$

$$
= \begin{cases}
0 & \textit{if } a \leq c \\
\frac{5c^{15}-210c^{13}+4368c^{11}-80080c^{9}-6435(a+1)^{5}\left(5a^{3}-25a^{2}+47a-35\right)c^{7}}{10543104}+ & \\
\frac{40040(a-2)(a+1)^{5}\left(5a^{3}-15a^{2}+18a-4\right)c^{6}}{10543104}- & \\
\frac{18018(a+1)^{5}\left(30a^{5}-150a^{4}+285a^{3}-225a^{2}+17a+35\right)c^{5}}{10543104}+ & \\
\frac{10920\left(75a^{11}-440a^{9}+1122a^{7}-1848a^{5}-660a^{4}+1155a^{3}+792a^{2}-68\right)c^{4}}{10543104}- & \\
\frac{30030\left(25a^{12}-162a^{10}+459a^{8}-812a^{6}-192a^{5}+735a^{4}+384a^{3}-210a^{2}-192a-35\right)c^{3}}{10543104}+ & \\
\frac{840\left(495a^{13}-3627a^{11}+11726a^{9}-23166a^{7}-3432a^{6}+27027a^{5}+10296a^{4}-15015a^{3}-10296a^{2}+872\right)c^{2}}{10543104}- & \\
\frac{5148\left(25a^{14}-210a^{12}+791a^{10}-1820a^{8}-160a^{7}+2695a^{6}+672a^{5}-2450a^{4}-1120a^{3}+1225a^{2}+1120a+256\right)c}{10543104}+ & \\
\frac{17160a^{15}-166320a^{13}+740376a^{11}-2082080a^{9}-205920a^{8}+3963960a^{7}+1153152a^{6}-5045040a^{5}-2882880a^{4}}{10543104}+ & \\
\frac{4204200a^{3}+5765760a^{2}+2635776a+437920}{10543104} & \textit{if } a \in [c \\
\frac{5c^{15}-210c^{13}+4368c^{11}-80080c^{9}+1647360c^{7}-5125120c^{6}+4612608c^{5}}{(256)(41184)}+ & \\
\frac{1397760c^{4}-4300800c^{2}-5271552c+8536064}{(256)(41184)}+ & \\
\frac{(40a-40)c^{7}}{256}+\frac{(140-140a^{2})c^{6}}{256}+\frac{\left(280a^{3}-168a-112\right)c^{5}}{256}+ & \\
\frac{\left(-350a^{4}-420a^{2}-70\right)c^{4}+\left(280a^{5}-560a^{3}+280a\right)c^{3}}{256}+ & \\
\frac{\left(-140a^{6}+420a^{4}-420a^{2}+140\right)c^{2}}{256}+ & \\
\frac{\left(40a^{7}-168a^{5}+280a^{3}-280a+128\right)c-5a^{8}+28a^{6}-70a^{4}+140a^{2}+128a-221}{256} & \textit{if } a \in [1 \\
\frac{5c^{15}-210c^{13}+4368c^{11}-80080c^{9}+205920c^{8}-512512c^{6}}{10543104}+ & \\
\frac{1397760c^{4}-4300800c^{2}-15814656c-2007040+10543104a}{10543104} & \textit{if } a \geq c
\end{cases}
$$

2. *For* $c \in [-2, 0]$

$$= \begin{cases} 0 & \textit{if } a \leq -1 \\ -\frac{(a+1)^5\left(6435\left(5a^3-25a^2+47a-35\right)c^7-40040(a-2)\left(5a^3-15a^2+18a-4\right)c^6\right.}{10543104}+ & \\ \frac{18018\left(30a^5-150a^4+285a^3-225a^2+17a+35\right)c^5}{10543104}- & \\ \frac{10920\left(75a^6-375a^5+685a^4-425a^3-228a^2+340a-68\right)c^4}{10543104}+ & \\ \frac{30030(a-1)^2(a+1)\left(25a^4-100a^3+138a^2-52a-35\right)c^3}{10543104}- & \\ \frac{840(a+1)^2\left(495a^6-3465a^5+10233a^4-16191a^3+14120a^2-6104a+872\right)c^2}{10543104}+ & \\ \frac{5148(a+1)^3\left(5a^3-20a^2+29a-16\right)^2c-17160a^{10}+85800a^9-91080a^8-231000a^7}{10543104}+ & \\ \frac{\left.553224a^6+42840a^5-984760a^4+415160a^3+844320a^2-446176a-437920\right)}{10543104} & \textit{if } a \in [-1, c+1] \\ \frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^9-512512c^6+1397760c^4}{10543104}+ & \\ \frac{-4300800c^2-5271552c+8536064-2882880a^4+5765760a^2+5271552a}{10543104} & \textit{if } a \in [c+1, 1] \\ \frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^9+205920c^8-512512c^6+1397760c^4}{10543104}+ & \\ \frac{-4300800c^2-5271552c-2007040+10543104a}{10543104} & \textit{if } a \geq 1 \end{cases}$$

3. *For* $c \geq 2$

$$= \begin{cases} \frac{-(c-a-1)^5((c-a)(5(c-a)(c-a+5)+47)+35)}{256} & \textit{if } a \in [c-1, c+1] \\ a-c & \textit{if } a \geq c+1 \end{cases}$$

4. *For* $c \leq -2$

$$= \begin{cases} -\frac{5a^8-28a^6+70a^4-140a^2-128a-35}{256} & \textit{if } a \in [-1, 1] \\ a & \textit{if } a \geq 1 \end{cases}$$

Proof. Let K be a Triweight kernel as in Def B.5.1. The CDF is defined in B.5.1. The partial L2-product of CDF as in Eqn (5.4.3). There are several things to consider in computing $\gamma_K(a, c)$ for Triweight kernel. For $c \geq 0$, the intersection of two Triweight CDFs occurs between $c - 1$ until ∞, whereas for $c \leq 0$ the intersection occurs between $(0, \infty]$. Therefore, we need to consider these cases: (1) $c \in [0, 2]$; (2) $[-2, 0]$; (3) $c \geq 2$; (4) $c \leq -2$.

i. **Case 1:**$c \in [0, 2]$**:** We need to consider the 4 cases below.

a. Case 1(a) $a \leq c-1$

$$\gamma_K(a,c) = 0 \tag{B.5.14}$$

b. Case 1(b) $a \in [c-1, 1]$:

$$\begin{aligned}
&\gamma_K(a,c)\\
&\quad=\int_{c-1}^{a} \frac{1}{32^2}\left(-5t^7+21t^5-35t^3+35t+16\right)\times\\
&\quad\left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\ dt &&\text{(B.5.15)}\\
&\quad=\frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9-6435\,(a+1)^5\left(5a^3-25a^2+47a-35\right)c^7}{10543104}+\\
&\quad\frac{40040\,(a-2)\,(a+1)^5\left(5a^3-15a^2+18a-4\right)c^6}{10543104}-\\
&\quad\frac{18018\,(a+1)^5\left(30a^5-150a^4+285a^3-225a^2+17a+35\right)c^5}{10543104}+\\
&\quad\frac{10920\left(75a^{11}-440a^9+1122a^7-1848a^5-660a^4+1155a^3+792a^2-68\right)c^4}{10543104}-\\
&\quad\frac{30030\left(25a^{12}-162a^{10}+459a^8-812a^6-192a^5+735a^4+384a^3-210a^2-192a-35\right)c^3}{10543104}+\\
&\quad\frac{840\left(495a^{13}-3627a^{11}+11726a^9-23166a^7-3432a^6+27027a^5+10296a^4-15015a^3-10296a^2+872\right)c^2}{10543104}-\\
&\quad\frac{5148\left(25a^{14}-210a^{12}+791a^{10}-1820a^8-160a^7+2695a^6+672a^5-2450a^4-1120a^3+1225a^2+1120a+256\right)c}{10543104}+\\
&\quad\frac{17160a^{15}-166320a^{13}+740376a^{11}-2082080a^9-205920a^8+3963960a^7+1153152a^6-5045040a^5-2882880a^4}{10543104}+\\
&\quad\frac{4204200a^3+5765760a^2+2635776a+437920}{10543104} &&\text{(B.5.16)}
\end{aligned}$$

Check: When $a=1$

$$\begin{aligned}
&=\frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9+1647360c^7-5125120c^6+4612608c^5}{10543104}+\\
&\frac{1397760c^4-4300800c^2-5271552c+8536064}{10543104} &&\text{(B.5.17)}
\end{aligned}$$

c. Case 1(c) $a \in [1, c+1]$

$$\begin{aligned}
\gamma_K(a,c) =& \int_{c-1}^{1} \frac{1}{32^2}\left(-5t^7+21t^5-35t^3+35t+16\right)\times \\
& \left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\,dt+ && \text{(B.5.18)}\\
& \int_{1}^{a} \frac{1}{32}\left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\,dt && \text{(B.5.19)}\\
=& \frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9+1647360c^7-5125120c^6+4612608c^5}{(256)(41184)}+ \\
& \frac{1397760c^4-4300800c^2-5271552c+8536064}{(256)(41184)}+ \\
& \frac{(40a-40)c^7}{256}+\frac{(140-140a^2)c^6}{256}+\frac{\left(280a^3-168a-112\right)c^5}{256}+ \\
& \frac{\left(-350a^4-420a^2-70\right)c^4+\left(280a^5-560a^3+280a\right)c^3}{256}+ \\
& \frac{\left(-140a^6+420a^4-420a^2+140\right)c^2}{256}+ \\
& \frac{\left(40a^7-168a^5+280a^3-280a+128\right)c-5a^8+28a^6-70a^4+140a^2+128a-221}{256} \\
& && \text{(B.5.20)}
\end{aligned}$$

d. Case 1(d) $a \geq c+1$

$$\begin{aligned}
\gamma_K(a,c) =& \int_{c-1}^{1} \frac{1}{32^2}\left(-5t^7+21t^5-35t^3+35t+16\right)\times \\
& \left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\,dt+ \\
& \int_{1}^{c+1} \frac{1}{32}\left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\,dt+ \\
& \int_{c+1}^{a} 1\,dt && \text{(B.5.21)}\\
=& \frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9+205920c^8-512512c^6}{10543104}+ \\
& \frac{1397760c^4-4300800c^2-15814656c-2007040+10543104a}{10543104} && \text{(B.5.22)}
\end{aligned}$$

ii. **Case 2:** $c \leq 0$ Under this case, the intersection between the two CDF occurs in the region $[-1, \infty]$

a. Case 2(a) $a \leq -1$:

$$\gamma_K(a,c) = 0 \qquad \text{(B.5.23)}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}
&\gamma_K(a,c)\\
&=\int_{-1}^{a} \frac{1}{32^2}\left(-5x^7+21x^5-35x^3+35x+16\right)\times\\
&\left(-5(x-c)^7+21(x-c)^5-35(x-c)^3+35(x-c)+16\right)\ dx
\end{aligned}\tag{B.5.24}$$

$$\begin{aligned}
&=-\ \frac{(a+1)^5\left(6435\left(5a^3-25a^2+47a-35\right)c^7-40040\,(a-2)\left(5a^3-15a^2+18a-4\right)c^6\right.}{10543104}+\\
&\frac{18018\left(30a^5-150a^4+285a^3-225a^2+17a+35\right)c^5}{10543104}-\\
&\frac{10920\left(75a^6-375a^5+685a^4-425a^3-228a^2+340a-68\right)c^4}{10543104}+\\
&\frac{30030\,(a-1)^2\,(a+1)\left(25a^4-100a^3+138a^2-52a-35\right)c^3}{10543104}-\\
&\frac{840\,(a+1)^2\left(495a^6-3465a^5+10233a^4-16191a^3+14120a^2-6104a+872\right)c^2}{10543104}+\\
&\frac{5148\,(a+1)^3\left(5a^3-20a^2+29a-16\right)^2c-17160a^{10}+85800a^9-91080a^8-231000a^7}{10543104}+\\
&\frac{\left.553224a^6+42840a^5-984760a^4+415160a^3+844320a^2-446176a-437920\right)}{10543104}
\end{aligned}\tag{B.5.25}$$

Check: When $a = c+1$,

$$\begin{aligned}
&=-\ \frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9+1647360c^7+5125120c^6+4612608c^5}{10543104}+\\
&\frac{1397760c^4-4300800c^2+5271552c+8536064}{10543104}
\end{aligned}\tag{B.5.26}$$

c. Case 2(c) $a \in [c+1, 1]$:

$$\begin{aligned}
\gamma_K(a,c)=&\int_{-1}^{c+1}\frac{1}{32^2}\left(-5t^7+21t^5-35t^3+35t+16\right)\times\\
&\left(-5(t-c)^7+21(t-c)^5-35(t-c)^3+35(t-c)+16\right)\ dt+\\
&\int_{c+1}^{a}\frac{1}{32}\left(-5t^7+21t^5-35t^3+35t+16\right)\ dt\\
=&\frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^9-512512c^6+1397760c^4}{10543104}+\\
&\frac{-4300800c^2-5271552c+8536064-2882880a^4+5765760a^2+5271552a}{10543104}
\end{aligned}\tag{B.5.27}$$

Check: When $a = 1$,

$$=\frac{-5c^{15} + 210c^{13} - 4368c^{11} + 80080c^{9} + 205920c^{8} - 512512c^{6}}{10543104} + \frac{1397760c^{4} - 4300800c^{2} - 5271552c + 8536064}{10543104} \tag{B.5.28}$$

d. Case 2(d) $a \geq 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{1}{32^2}\left(-5t^7 + 21t^5 - 35t^3 + 35t + 16\right) \times \\ &\left(-5(t-c)^7 + 21(t-c)^5 - 35(t-c)^3 + 35(t-c) + 16\right)\, dt + \\ &\int_{c+1}^{1} \frac{1}{32}\left(-5t^7 + 21t^5 - 35t^3 + 35t + 16\right)\, dt + \\ &\int_{1}^{a} 1\, dt \\ =& \frac{-5c^{15} + 210c^{13} - 4368c^{11} + 80080c^{9} + 205920c^{8} - 512512c^{6} + 1397760c^{4}}{10543104} + \\ &\frac{-4300800c^{2} - 5271552c - 2007040 + 10543104a}{10543104}\end{aligned}$$

iii. **Case 3:** $c \geq 2$: We need to consider the cases below.

a. Case 3(a) $a \in [c-1, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{a} \left(-5(t-c)^7 + 21(t-c)^5 - 35(t-c)^3 + 35(t-c) + 16\right)\, dt \\ =& \frac{-\left(c-a-1\right)^5\left(\left(c-a\right)\left(5\left(c-a\right)\left(c-a+5\right)+47\right)+35\right)}{256}\end{aligned} \tag{B.5.29}$$

b. Case 3(b) $a \geq 1$

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{c+1} \left(-5(t-c)^7 + 21(t-c)^5 - 35(t-c)^3 + 35(t-c) + 16\right)\, dt + \\ &\int_{c+1}^{a} 1\, dt \\ =& a - c\end{aligned} \tag{B.5.30}$$

iv. **Case 4:** $c \leq -2$: Under this condition, we consider the sub-cases below.

a. Case 4(a) $a \in [-1, 1]$

$$\begin{aligned}\gamma_K(a,c) =& \int \left(-5t^7 + 21t^5 - 35t^3 + 35t + 16\right)\ dt \\ =& -\frac{5a^8 - 28a^6 + 70a^4 - 140a^2 - 128a - 35}{256}\end{aligned} \tag{B.5.31}$$

b. Case 4(b) $a \geq 1$

$$\begin{aligned}\gamma_K(a,c) =& \int \left(-5t^7 + 21t^5 - 35t^3 + 35t + 16\right)\ dt + \int_1^a 1\ dt \\ =& a\end{aligned} \tag{B.5.32}$$

□

Derivation B.5.4. *Let K be the Triweight kernel as in Def B.5.1. Let the CDF and the partial L2-product of the CDF as in B.5.1 and B.5.1, respectively. The partial L2-product of $1 - \nu_K$ at two different central points a and c for the integral boundary of $[a, \infty)$ where $a \in \mathbb{R}$ is*

1. For $c \in [0,2]$

$$
= \begin{cases}
0 & \textit{if } a \geq 1 \\
\frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^{9}+6435(-a+1)^{5}\left(-5a^{3}-25a^{2}-47a-35\right)c^{7}}{10543104}+ & \\
\frac{40040(-a-2)(-a+1)^{5}\left(-5a^{3}-15a^{2}-18a-4\right)c^{6}}{10543104}+ & \\
\frac{18018(-a+1)^{5}\left(-30a^{5}-150a^{4}-285a^{3}-225a^{2}-17a+35\right)c^{5}}{10543104}+ & \\
\frac{10920\left(-75a^{11}+440a^{9}-1122a^{7}+1848a^{5}-660a^{4}-1155a^{3}+792a^{2}-68\right)c^{4}}{10543104}+ & \\
\frac{30030\left(25a^{12}-162a^{10}+459a^{8}-812a^{6}+192a^{5}+735a^{4}-384a^{3}-210a^{2}+192a-35\right)c^{3}}{10543104}+ & \\
\frac{840\left(-495a^{13}+3627a^{11}-11726a^{9}+23166a^{7}-3432a^{6}-27027a^{5}+10296a^{4}+15015a^{3}-10296a^{2}+872\right)c^{2}}{10543104}+ & \\
\frac{5148\left(25a^{14}-210a^{12}+791a^{10}-1820a^{8}+160a^{7}+2695a^{6}-672a^{5}-2450a^{4}+1120a^{3}+1225a^{2}-1120a+256\right)c}{10543104}+ & \\
\frac{-17160a^{15}+166320a^{13}-740376a^{11}+2082080a^{9}-205920a^{8}-3963960a^{7}+1153152a^{6}+5045040a^{5}-2882880a^{4}}{10543104}+ & \\
\frac{-4204200a^{3}+5765760a^{2}-2635776a+437920}{10543104} & \textit{if } a \in [c-1,1] \\
\frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^{9}-1647360c^{7}-5125120c^{6}-4612608c^{5}}{(256)(41184)}+ & \\
\frac{1397760c^{4}-4300800c^{2}+5271552c+8536064}{(256)(41184)}+ & \\
\frac{(-40a-40)c^{7}}{256}+\frac{(140-140a^{2})c^{6}}{256}+\frac{\left(-280a^{3}+168a-112\right)c^{5}}{256}+ & \\
\frac{\left(-350a^{4}-420a^{2}-70\right)c^{4}+\left(-280a^{5}+560a^{3}-280a\right)c^{3}}{256}+ & \\
\frac{\left(-140a^{6}+420a^{4}-420a^{2}+140\right)c^{2}}{256}+ & \\
\frac{\left(-40a^{7}+168a^{5}-280a^{3}+280a+128\right)c-5a^{8}+28a^{6}-70a^{4}+140a^{2}-128a-221}{256} & \textit{if } a \in [-1,c-1] \\
\frac{-5c^{15}+210c^{13}-4368c^{11}+80080c^{9}+205920c^{8}-512512c^{6}}{10543104}+ & \\
\frac{1397760c^{4}-4300800c^{2}+15814656c-2007040+10543104a}{10543104} & \textit{if } a \leq -1
\end{cases}
$$

2. *For* $c \in [-2, 0]$

$$= \begin{cases} 0 & \textit{if } ac+1 \\ -\frac{(-a+1)^5(-(6435(-5a^3-25a^2-47a-35))c^7-40040(-a-2)(-5a^3-15a^2-18a-4)c^6}{10543104} - & \\ \frac{18018(-30a^5-150a^4-285a^3-225a^2-17a+35)c^5}{10543104} - & \\ \frac{10920(75a^6+375a^5+685a^4+425a^3-228a^2-340a-68)c^4}{10543104} - & \\ \frac{30030(-a-1)^2(-a+1)(25a^4+100a^3+138a^2+52a-35)c^3}{10543104} - & \\ \frac{840(-a+1)^2(495a^6+3465a^5+10233a^4+16191a^3+14120a^2+6104a+872)c^2}{10543104} - & \\ \frac{5148(-a+1)^3(-5a^3-20a^2-29a-16)^2c-17160a^{10}-85800a^9-91080a^8+231000a^7}{10543104} + & \\ \frac{553224a^6-42840a^5-984760a^4-415160a^3+844320a^2+446176a-437920)}{10543104} & \textit{if } a \in [-1, c+1] \\ \frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9-512512c^6+1397760c^4}{10543104} + & \\ \frac{-4300800c^2-5271552c+8536064-2882880a^4+5765760a^2+5271552a}{10543104} & \textit{if } a \in [c-1, -1] \\ \frac{5c^{15}-210c^{13}+4368c^{11}-80080c^9+205920c^8-512512c^6+1397760c^4}{10543104} + & \\ \frac{-4300800c^2+5271552c-2007040-10543104a}{10543104} & \textit{if } a \leq c-1 \end{cases}$$

3. *For* $c \geq 2$

$$= \begin{cases} -\frac{(c-a+1)^5((c-a)(5(c-a-5)(c-a)+47)-35)}{256} & \textit{if } a \in [c-1, c+1] \\ c-a & \textit{if } a \leq c-1 \end{cases}$$

4. *For* $c \leq -2$

$$= \begin{cases} \frac{5a^8-28a^6+70a^4-140a^2-128a-35}{256} & \textit{if } a \in [-1,1] \\ -a & \textit{if } a \leq 1 \end{cases}$$

B.6 Triangle Kernel

Definition B.6.1. *Let a triangle kernel be defined as*

$$K(u) = \begin{cases} 1 - |u| & \textit{if} -1 \leq u \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

So, we have

$$K(u) = \begin{cases} 1 - u & \textit{if}\, 0 < u \leq 1 \\ 1 + u & \textit{if } -1 \leq u < 0 \\ 0 & \textit{otherwise} \end{cases}$$

Derivation B.6.1. *Let K be a Triangle kernel as defined as in Def B.6.1. The CDF of the kernel is*

$$\nu_K(a) = \begin{cases} \frac{1+2a+a^2}{2} & \textit{if} \quad a \leq 0 \\ \frac{1+2a-a^2}{2} & \textit{if} \quad a \geq 0 \end{cases}$$

Proof. There are two cases that we need to consider, $a \in [0, 1]$ and $a \in [-1, 0]$.

1. **Case 1:** $a \in [-1, 0]$

$$\nu_K(a) = \int_{-\infty}^{a} (1 - |u|)\mathbb{1}(u \in [-1, 1])\, du = \int_{-1}^{a} (1 + u)\, du$$
$$= \frac{a^2}{2} + \frac{1}{2} + a$$

2. **Case 2:** $a \in [0, 1]$

$$\nu_K(a) = \int_{-1}^{0} (1 + u)\, du + \int_{0}^{a} (1 - u)\, du = \frac{1}{2} + a - \frac{a^2}{2} \tag{B.6.1}$$

□

Derivation B.6.2. *Let K be a Triangle kernel in Def B.6.1. Then, the partial L2-product of the Triangle kernel at two central points 0 and $c \in \mathbb{R}$, is*

1. For $c \geq 0$

$$\lambda_K(c) = \begin{cases} 0 & \textit{if} \quad |c| > 2 \\ \frac{-c^3+6c^2-12c+8}{6} & \textit{if} \quad c \in [1, 2] \\ \frac{3c^3-6c^2+4}{6} & \textit{if} \quad c \in [0, 1] \end{cases}$$

2. *For* $c \leq 0$

$$\lambda_K(c) = \begin{cases} 0 & \textit{if} \quad |c| < 2 \\ \frac{c^3+6c^2+12c+8}{6} & \textit{if} \quad c \in [-2,-1] \\ \frac{-3c^3+6c^2+4}{6} & \textit{if} \quad c \in [-1,0] \end{cases}$$

Proof. Suppose we have the Triangular kernel as in Def B.6.1. Then, the L2-norm of the kernel as in Eqn (5.4.3) for Triangular kernel is

$$\lambda_K(c) = \int (1-|u|)(1-|u|)\mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [-1,1])\, du. \tag{B.6.2}$$

There are two cases to consider: (1) $c \geq 0$; (2) $c \leq 0$.

1. **Case 1:** $c \geq 0$**:** There are three cases to be considered under this case.
   a. Case 1(a) $c \geq 2$:

$$\lambda_K(c) = 0 \tag{B.6.3}$$

   b. Case 1(b) $c \in [0,1]$:

$$\begin{aligned}\lambda_K(c) =& \int_{c-1}^{0} (1+u)\,(1+(u-c))\; du + \int_{0}^{c} (1-u)\,(1+(u-c))\; du + \\ & \int_{c}^{1} (1-u)\,(1-(u-c))\; du \\ =& \frac{3c^3-6c^2+4}{6}\end{aligned} \tag{B.6.4}$$

   c. Case 1(c) $c \in [1,2]$:

$$\begin{aligned}\lambda_K(c) =& \int_{c-1}^{1} ((1-c)+cu-u^2)\; du \\ =& \frac{-c^3+6c^2-12c+8}{6}\end{aligned} \tag{B.6.5}$$

2. **Case 2:** $c \leq 0$**:** There are three cases to be considered.
   a. Case 2(a) $c \leq -2$:

$$\lambda_K(c) = 0$$

b. Case 2(b): $c \in [-1, 0]$:

$$\begin{aligned}\lambda_K(c) =& \int_{-1}^{c} (1+u)\,(1+(u-c))\ du + \int_{c}^{0} (1+u)\,(1-(u-c))\ du + \\ & \int_{0}^{c+1} (1-u)\,(1-(u-c))\ du \\ =& \frac{-3c^3+6c^2+4}{6}\end{aligned} \tag{B.6.6}$$

c. Case 2(b): $c \in [-2, -1]$

$$\lambda_K(c) = \int_{-1}^{c+1} (1+u)\,(1-(u-c))\ du = \frac{c^3+6c^2+12c+8}{6} \tag{B.6.7}$$

□

Derivation B.6.3. *Let K be a Triangle kernel as in Def B.6.1. Then, the partial L2-product of the Triangle kernel at two central points 0 and $c \in \mathbb{R}$, from $-\infty$ to $a \in \mathbb{R}$ is*

1. *For $c \in [1, 2]$*

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \le c-1 \\ \frac{-c^3+6c^2-\left(-3a^2+6a+9\right)c-2a^3+6a+4}{6} & \textit{if } a \in [c-1, 1] \end{cases}$$

2. *For $c \in [0, 1]$*

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \le c-1 \\ \frac{c^3+\left(-3a^2-6a-3\right)c+2a^3+6a^2+6a+2}{6} & \textit{if } a \in [c-1, 0] \\ \frac{(-2a^3+3a^2c-6ac+6a+c^3-3c+2)}{6} & \textit{if } a \in [0, c] \\ \frac{(2a^3-3a^2c-6a^2+6ac+6a+3c^3-6c^2-3c+2)}{6} & \textit{if } a \in [c, 1] \end{cases}$$

3. *For $c \in [-1, 0]$*

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \le -1 \\ \frac{-(3a^2+6a+3)c+2a^3+6a^2+6a+2}{6} & \textit{if } a \in [-1, c] \\ \frac{-2a^3+3a^2+12a-2c^3-6c^2+3c-4}{6} & \textit{if } a \in [c, 0] \\ \frac{2a^3-3a^2c-6a^2+6ac+6a-2c^3-6c^2-3c+2}{6} & \textit{if } a \in [0, c+1] \end{cases}$$

4. *For* $c \in [-2,-1]$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{(3a^2+6a)c-2a^3+6a+3c+4}{6} & \textit{if } a \in [-1, c+1] \end{cases}$$

Proof. Suppose we have the Triangle kernel as in Def B.6.1. Then, the partial L2-product of the kernel K as in Eqn (5.4.3) for Triangle kernel is

$$\lambda_K(a,c) = \int_{-\infty}^{a} (1-|u|)(1-|(u-c)|)\mathbb{1}(u \in [-1,1])\mathbb{1}(u \in [c-1,c+1])\, du$$

There are 4 cases to consider: (1) $c \in [0,1]$; (2) $c \in [1,2]$; (3) $c \in [-1,0]$; (4) $c \in [-2,-1]$.

1. **Case 1:** $c \in [1,2]$
   a. Case 1(a) $a \leq c-1$:

$$\lambda_K(a,c) = \int_{c-1}^{a} (1-|u|)(1-|u-c|)\, du = 0. \tag{B.6.8}$$

   b. Case 1b $a \in [c-1, a]$:

$$\begin{aligned}\lambda_K(a,c) &= \int_{c-1}^{a} (1-u)(1+(u-c))\, du \\ &= \frac{-c^3+6c^2-(-3a^2+6a+9)\,c-2a^3+6a+4}{6}\end{aligned} \tag{B.6.9}$$

2. **Case 2** $c \in [0,1]$**:** The intersection of both kernels is still in the region $[c-1,1]$. However, the direction of the function changes in between that region.
   a. Case 2(a) $a \leq c-1$:

$$\lambda_K(a,c) \int_{c-1}^{c-1} (1-|u|)(1-|u-c|)\, du = 0 \tag{B.6.10}$$

   b. Case 2(b) $a \in [c-1, 0]$:

$$\begin{aligned}\lambda_K(a,c) &= \int_{c-1}^{a} (1+u)(1+(u-c))\, du \\ &= \frac{c^3+(-3a^2-6a-3)\,c+2a^3+6a^2+6a+2}{6}\end{aligned} \tag{B.6.11}$$

c. Case 2(c) $a \in [0, c]$:

$$\begin{aligned}\lambda_K(a,c) =& \int_{c-1}^{0} (1+u)(1+(u-c))\, du + \int_{0}^{a} (1-u)(1+(u-c))\, du \\ =& \frac{(-2a^3+3a^2c-6ac+6a+c^3-3c+2)}{6} \qquad \text{(B.6.12)}\end{aligned}$$

d. Case 2(d) $a \in [c, 1]$:

$$\begin{aligned}\lambda_K(a,c) =& \int_{c-1}^{0} (1+u)(1+(u-c))\, du + \int_{0}^{c} (1-u)(1+(u-c))\, du + \\ & \int_{c}^{a} (1-u)(1-(u-c))\, du \\ =& \frac{(2a^3-3a^2c-6a^2+6ac+6a+3c^3-6c^2-3c+2)}{6} \qquad \text{(B.6.13)}\end{aligned}$$

3. **Case 3:** $c \in [-1, 0]$**:** When $c \in [-1, 0]$, the intersection region of the two kernels is $-[-1, c+1]$

a. Case 3(a) $a \leq -1$:

$$\lambda_K(a,c) = \int_{-1}^{-1} (1+u)(1+(u-c))du = 0 \qquad \text{(B.6.14)}$$

b. Case 3(b) $a \in [-1, c]$:

$$\begin{aligned}\lambda_K(a,c) =& \int_{-1}^{a} (1+u)(1+(u-c))du \\ =& \frac{-(3a^2+6a)c+2a^3+6a^2+6a+3c+2}{6} \qquad \text{(B.6.15)}\end{aligned}$$

c. Case 3(c) $a \in [c, 0]$:

$$\begin{aligned}\lambda_K(a,c) =& \int_{-1}^{c} (1+u)(1+(u-c))\, du + \int_{c}^{a} (1+u)(1-(u-c))\, du \\ =& \frac{-2a^3+3a^2c+6a+6ac-2c^3-6c^2+3c-4}{6} \qquad \text{(B.6.16)}\end{aligned}$$

d. Case 3(d) $a \in [0, c+1]$:

$$\lambda_K(a,c) = \int_{-1}^{c} (1+u)(1+(u-c))\,du + \int_{c}^{0} (1+u)(1-(u-c))\,dx + \int_{0}^{a} (1-u)(1-(u-c))\,du$$
$$= \frac{2a^3 - 3a^2c - 6a^2 + 6ac + 6a - 2c^3 - 6c^2 - 3c + 2}{6} \tag{B.6.17}$$

4. **Case 4:** $c \in [-2, -1]$ Under this condition, the intersection region of the two kernels is $[-1, c+1]$

(a) Case 4(a) $a \leq -1$:

$$\lambda_K(a,c) = \int_{-1}^{-1} (1-|u|)(1-|u-c|)\,du = 0 \tag{B.6.18}$$

(b) Case 4(b) $a \in [-1, c+1]$:

$$\lambda_K(a,c) = \int_{-1}^{a} (1-(u-c))(1+u)\,du$$
$$= \frac{(3a^2+6a)c - 2a^3 + 6a + 3c + 4}{6} \tag{B.6.19}$$

□

Derivation B.6.4. *Let K be a Triangle kernel as in Def B.6.1. The partial L2-product of Triangle CDF with different centre points, 0 and $c \in \mathbb{R}$ for the integral boundary $-\infty$ to $a \in \mathbb{R}$,*

1. For $c \leq -2$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} a \leq -1 \\ \frac{a^3+3a^2+3a+1}{6} & \textit{if } a \in [-1,0] \\ \frac{-a^3+3a^2+3a+1}{6} & \textit{if } a \in [0,1] \\ a & \textit{if } a \geq 1 \end{cases}$$

2. *For* $c \in [-2,-1]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{-10c^2-25c+4}{120} + \frac{a^3\left(-10c^2+20c+40\right)}{120} + \\ \frac{a^2\left(-30c^2-30c+60\right)}{120} + \frac{a\left(-30c^2-60c+30\right)}{120} + \frac{a^4c}{8} - \frac{a^5}{20} & \textit{if } a \in [-1, c+1] \\ \frac{-c^5-10c^4-40c^3-80c^2-80c+20a^3+60a^2+60a-12}{120} & \textit{if } a \in [c+1, 0] \\ \frac{-c^5-10c^4-40c^3-80c^2-80c-20a^3+10a^2+60a}{120} & \textit{if } a \in [0,1] \\ \frac{-c^5-10c^4-40c^3-80c^2-80c+120a-32}{120} & \textit{if } a \geq 1 \end{cases}$$

3. *For* $c \in [-1,0]$

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{a\left(30c^2-60c+30\right)}{120} + \frac{a^2\left(30c^2-90c+60\right)}{120} + \frac{10c^2-15c+6}{120} + \\ \frac{a^3\left(10c^2-60c+60\right)}{120} + \frac{a^4(30-15c)}{120} + \frac{a^5}{20} & \textit{if } a \in [-1, c] \\ \frac{c^5+10c^4+20c^3-30c}{120} + \frac{c^5}{120} + \frac{c^2}{12} + \frac{a^3\left(-10c^2+20c+40\right)}{120} + \\ \frac{a\left(-30c^2-60c+30\right)}{120} + \frac{a^2\left(-30c^2-30c+60\right)}{120} + \frac{a^4c}{8} + \frac{c}{8} - \frac{a^5}{20} + \\ \frac{1}{20} & \textit{if } a \in [c, 0] \\ \frac{c^5+10c^2+15c+6}{120} + \frac{c\left(c^4+10c^3+20c^2-30\right)}{120} + \frac{\left(10a^3-30a^2-30a\right)c^2}{120} + \\ \frac{\left(-15a^4+60a^3-30a^2-60a\right)c}{120} + \frac{6a^5-30a^4+20a^3+60a^2+30a}{120} & \textit{if } a \in [0, c+1] \\ \frac{3c^5+10c^4-40c^2-60c-20a\left(a^2-3a-3\right)-8}{120} & \textit{if } a \in [c+1, 1] \\ \frac{3c^5+10c^4-40c^2-60c+120a-28}{120} & \textit{if } a \geq 1 \end{cases}$$

4. *For* $c \in [0,1]$

$$\gamma_K(a,c)=\begin{cases} 0 & \textit{if } a \le c-1 \\ \frac{-c^5+10c^2-15c+6}{120}+\frac{a\left(30c^2-60c+30\right)}{120}+\frac{a^2\left(30c^2-90c+60\right)}{120}+ & \\ \frac{a^3\left(10c^2-60c+60\right)}{120}+\frac{a^4(30-15c)}{120}+\frac{a^5}{20} & \textit{if } a \in [c-1,0] \\ -\frac{c^5}{120}+\frac{c^2}{12}-\frac{c}{8}+\frac{a\left(30c^2-60c+30\right)}{120}+\frac{a^2\left(30c^2-90c+60\right)}{120}+ & \\ \frac{a^3\left(-10c^2-20c+40\right)}{120}+\frac{a^4c}{8}-\frac{a^5}{20}+\frac{1}{20} & \textit{if } a \in [0,c] \\ -\frac{c^5}{60}+\frac{-c^5+40c^3-30c}{120}+\frac{c^4}{12}-\frac{c^3}{6}+\frac{c^2}{12}+\frac{a\left(-30c^2-60c+30\right)}{120}+ & \\ \frac{a^2\left(-30c^2-30c+60\right)}{120}+\frac{a^3\left(10c^2+60c+20\right)}{120}+\frac{c}{8}+\frac{a^4(-15c-30)}{120}+ & \\ \frac{a^5}{20}+\frac{1}{20} & \textit{if } a \in [c,1] \\ \frac{-3c^5+10c^4+20c^3+20c^2-8}{120}+\frac{a\left(-60c^2-120c+60\right)}{120}+\frac{a^2(60c+60)}{120}- & \\ \frac{a^3}{6} & \textit{if } a \in [1,c+1] \\ \frac{-3c^5+10c^4-40c^2-60c+120a-28}{120} & \textit{if } a \ge c+1 \end{cases}$$

5. *For* $c \in [1,2]$

$$\gamma_K(a,c)=\begin{cases} 0 & \textit{if } a \le c-1 \\ \frac{c^5-10c^4+20c^3-10c^2-5c+4}{120}+\frac{a\left(30c^2-60c+30\right)}{120}+ & \\ \frac{a^2\left(30c^2-90c+60\right)}{120}+\frac{a^3\left(-10c^2-20c+40\right)}{120}+\frac{a^4c}{8}-\frac{a^5}{20} & \textit{if } a \in [c-1,1] \\ \frac{c^5-10c^4+20c^3-20c^2+20c-60a^2(c-1)+60a(c-1)^2+20a^3-12}{120} & \textit{if } a \in [1,c] \\ \frac{c^5-10c^4+60c^3}{120}+\frac{(-60a-20)c^2}{120}+\frac{\left(60a^2-120a+20\right)c}{120}+ & \\ \frac{-20a^3+60a^2+60a-12}{120} & \textit{if } a \in [c,c+1] \\ \frac{c^5-10c^4+40c^3-80c^2-40c+120a-32}{120} & \textit{if } a \ge c+1 \end{cases}$$

6. *For* $c \ge 2:$

$$\gamma_K(a,c)=\begin{cases} 0 & \textit{if } a \le c-1 \\ \frac{(-c+a+1)^3}{120} & \textit{if } a \in [c-1,c] \\ \frac{c^3+(3-3a)c^2+\left(3a^2-6a-3\right)c-a^3+3a^2+3a+1}{6} & \textit{if } [c,c+1] \\ a-c & \textit{if } a \ge c+1 \end{cases}$$

Proof. Let K be a Triangle kernel as in Def B.6.1. The CDF is defined in B.6.1.

Then, the partial L2-product of Triangle CDF as in Eqn (5.4.3) is

$$\gamma_K(a,c) = \int_{-\infty}^{a} \frac{1+2t+t^2}{2}\frac{1+2(t-c)+(t-c)^2}{2}\mathbb{1}(t\in[-1,])\mathbb{1}(t\in[c-1,c+1])\,dt$$

1. **Case 1:** $c \leq -2$**:**
   a. Case 1(a) $a \leq -1$:

$$\gamma_K(a,c) = 0 \tag{B.6.20}$$

   b. Case 1(b) $a \in [-1,0]$:

$$\begin{aligned}\gamma_K(a,c) &= \int_{-1}^{a} \frac{t^2+2t+1}{2}\,dt \\ &= \frac{a^3+3a^2+3a+1}{6}\end{aligned} \tag{B.6.21}$$

   Check: When $a = 0$

$$\gamma_K(a,c) = \frac{1}{6} \tag{B.6.22}$$

   c. Case 1(c) $a \in [0,1]$:

$$\gamma_K(a,c) = \int_{-1}^{0} \frac{t^2+2t+1}{2}\,dt + \int_{0}^{a} \frac{-t^2+2t+1}{2}\,dt \tag{B.6.23}$$

$$= \frac{-a^3+3a^2+3a+1}{6} \tag{B.6.24}$$

   d. Case 1(d) $a \geq 1$:

$$\gamma_K(a,c) = \int_{-1}^{0} \frac{t^2+2t+1}{2}\,dt + \int_{0}^{1} \frac{-t^2+2t+1}{2}\,dt + \int_{1}^{a} 1\,dt = \quad a \tag{B.6.25}$$

2. **Case 2:** $c \in [-2,-1]$**:** The intersection of the two CDF in this region is $[-1,\infty]$. We consider the sub-cases below.
   a. Case 2(a) $a \leq -1$:

$$\gamma_K(a,c) = 0 \tag{B.6.26}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{a} \frac{-(t-c)^2 + 2(t-c) + 1}{2} \frac{t^2 + 2t + 1}{2}\, dt \\ =& \frac{-10c^2 - 25c + 4}{120} + \frac{a^3\left(-10c^2 + 20c + 40\right)}{120} + \\ & \frac{a^2\left(-30c^2 - 30c + 60\right)}{120} + \frac{a\left(-30c^2 - 60c + 30\right)}{120} + \\ & \frac{a^4 c}{8} - \frac{a^5}{20}\end{aligned} \tag{B.6.27}$$

Check: When $a = c + 1$

$$\gamma_K(a,c) = \frac{-c^5 - 10c^4 - 20c^3 + 40c^2 + 160c + 128}{120} \tag{B.6.28}$$

c. Case 2(c) $a \in [c+1, 0]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{-(t-c)^2 + 2(t-c) + 1}{2} \frac{t^2 + 2t + 1}{2}\, dt \\ & \int_{c+1}^{a} \frac{t^2 + 2t + 1}{2}\, dt \\ =& \frac{-c^5 - 10c^4 - 40c^3 - 80c^2 - 80c + 20a^3 + 60a^2 + 60a - 12}{120}\end{aligned} \tag{B.6.29}$$

check: When $a = 0$

$$\gamma_K(a,c) = \frac{-c^5 - 10c^4 - 40c^3 - 80c^2 - 80c - 12}{120} \tag{B.6.30}$$

d. Case 2(d) $a \in [0, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{-(t-c)^2 + 2(t-c) + 1}{2} \frac{t^2 + 2t + 1}{2}\, dt \\ & \int_{c+1}^{0} \frac{t^2 + 2t + 1}{2}\, dt + \int_{0}^{a} \frac{-t^2 + 2t + 1}{2}\, dt \\ =& \frac{-c^5 - 10c^4 - 40c^3 - 80c^2 - 80c - 20a^3 + 10a^2 + 60a}{120}\end{aligned} \tag{B.6.31}$$

check: When $a = 1$

$$\gamma_K(a,c) = \frac{-c^5 - 10c^4 - 40c^3 - 80c^2 - 80c + 88}{120} \tag{B.6.32}$$

e. Case 2(e) $a \geq 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{-(t-c)^2+2(t-c)+1}{2}\frac{t^2+2t+1}{2}\,dt\\ & \int_{c+1}^{0} \frac{t^2+2t+1}{2}\,dt + \int_{0}^{1} \frac{-t^2+2t+1}{2}\,dt + \int_{1}^{a} 1\,dt\\ =& \frac{-c^5-10c^4-40c^3-80c^2-80c+120a-32}{120} \end{aligned} \tag{B.6.33}$$

3. **Case 3:** $c \in [-1, 0]$**:** The intersection of the two CDF in this region is $[-1, \infty]$. There are multiple sub-cases to be considered.

a. Case 3(a) $a \in [-1, 0]$:

$$\gamma_K(a,c) = 0 \tag{B.6.34}$$

b. Case 3(b) $a \in [-1, c]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{a} (1+2t+t^2)(1+2(t-c)+(t-c)^2)\,dt\\ =& \frac{a\left(30c^2-60c+30\right)}{120} + \frac{a^2\left(30c^2-90c+60\right)}{120} + \frac{10c^2-15c+6}{120} +\\ & \frac{a^3\left(10c^2-60c+60\right)}{120} + \frac{a^4\left(30-15c\right)}{120} + \frac{a^5}{20} \end{aligned} \tag{B.6.35}$$

Check: When $a = c$

$$\gamma_K(a,c) = \frac{c^5+10c^2+15c+6}{120} \tag{B.6.36}$$

c. Case 3(c) $a \in [c, 0]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{c} (1+2t+t^2)(1+2(t-c)+(t-c)^2)\,dt +\\ & \frac{1}{4}\int_{c}^{a} (1+2t+t^2)(1+2(t-c)-(t-c)^2)\,dt\\ =& \frac{2c^5+10c^4+20c^3+c^2(-10a^3-30a^2-30a+10)}{120} +\\ & \frac{c(15a^4+20a^3-30a^2-60a-15)}{120} +\\ & \frac{-6a^5+40a^3+60a^2+30a+6}{120} \end{aligned} \tag{B.6.37}$$

Check: When $a = 0$

$$\gamma_K(a,c) = \frac{2c^5 + 10c^4 + 20c^3 + 10c^2 - 15c + 6}{120} \tag{B.6.38}$$

d. Case 3(d) $a \in [0, c+1]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{c} (1+2t+t^2)(1+2(t-c)+(t-c)^2)\,dt+\\ & \frac{1}{4}\int_{c}^{0} (1+2t+t^2)(1+2(t-c)-(t-c)^2)\,dt+\\ & \frac{1}{4}\int_{0}^{a} (1+2t-t^2)(1+2(t-c)-(t-c)^2)\,dt\\ =& \frac{c^5+10c^2+15c+6}{120} + \frac{c\,(c^4+10c^3+20c^2-30)}{120}+\\ & \frac{(10a^3-30a^2-30a)\,c^2+(-15a^4+60a^3-30a^2-60a)\,c}{120}+\\ & \frac{6a^5-30a^4+20a^3+60a^2+30a}{120}\end{aligned} \tag{B.6.39}$$

Check: When $a = 0$

$$\gamma_K(a,c) = \frac{2c^5 + 10c^4 + 20c^3 + 10c^2 - 15c + 6}{120} \tag{B.6.40}$$

Check: When $a = c + 1$

$$\gamma_K(a,c) = \frac{3c^5 + 10c^4 - 20c^3 - 40c^2 + 60c + 92}{120} \tag{B.6.41}$$

e. Case 3(e) $a \in [c+1, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{c} (1+2t+t^2)(1+2(t-c)+(t-c)^2)\,dt+\\ & \frac{1}{4}\int_{c}^{0} (1+2t+t^2)(1+2(t-c)-(t-c)^2)\,dt+\\ & \frac{1}{4}\int_{0}^{c+1} (1+2t-t^2)(1+2(t-c)-(t-c)^2)\,dt+\\ & \frac{1}{2}\int_{c+1}^{a} (1+2t-t^2)\,dt\\ =& \frac{-20a^3+60a^2+60a+3c^5+10c^4-40c^2-60c-8}{120}\\ & \frac{3c^5+10c^4-40c^2-60c-20a\,(a^2-3a-3)-8}{120}\end{aligned} \tag{B.6.42}$$

Check: When $a = c + 1$

$$\gamma_K(a,c) = \frac{3c^5 + 10c^4 - 20c^3 - 40c^2 + 60c + 92}{120} \tag{B.6.43}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{3c^5 + 10c^4 - 40c^2 - 60c + 92}{120} \tag{B.6.44}$$

f. Case 3(f) $a \geq 1$

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{c} (1 + 2t + t^2)(1 + 2(t-c) + (t-c)^2)\ dt+\\ & \frac{1}{4}\int_{c}^{0} (1 + 2t + t^2)(1 + 2(t-c) - (t-c)^2)\ dt+\\ & \frac{1}{4}\int_{0}^{c+1} (1 + 2t - t^2)(1 + 2(t-c) - (t-c)^2)\ dt+\\ & \frac{1}{2}\int_{c+1}^{1} (1 + 2t - t^2)\ dt + \int_{1}^{a} 1\ dt\\ =& \frac{3c^5 + 10c^4 - 40c^2 - 60c + 120a - 28}{120}\end{aligned} \tag{B.6.45}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{3c^5 + 10c^4 - 40c^2 - 60c + 92}{120} \tag{B.6.46}$$

4. **Case 4:** $c \in [0,1]$**:** The intersection of the two CDF is in the region $[c - 1, \infty]$.There are multiple considerations that we need to take,

a. Case 4(a) $a \leq c - 1$:

$$\gamma_K(a,c) = \int_{-\infty}^{a} \nu_K(t)\nu_K(t-c)\ dt = 0 \tag{B.6.47}$$

b. Case 4(b) $a \in [c - 1, 0]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{a} \left(\frac{(t-c)^2 + 2(t-c) + 1}{2}\right)\left(\frac{t^2 + 2t + 1}{2}\right)\ dt\\ =& \frac{-c^5 + 10c^2 - 15c + 6}{120} + \frac{a\left(30c^2 - 60c + 30\right)}{120} +\\ & \frac{a^2\left(30c^2 - 90c + 60\right)}{120} + \frac{a^3\left(10c^2 - 60c + 60\right)}{120} +\\ & \frac{a^4\left(30 - 15c\right)}{120} + \frac{a^5}{20}\end{aligned} \tag{B.6.48}$$

Check: When $a = c - 1$

$$\gamma_K(a,c) = \frac{((c-1)-c+1)^3(6(c-1)^2+3(c-1)(c+4)+c^2+3c+6)}{120} = 0 \tag{B.6.49}$$

Check: When $a = 0$,

$$\gamma_K(a,c) = \frac{-c^5+10c^2-15c+6}{120} \tag{B.6.50}$$

c. Case 4(c) $a \in [0,c]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{0} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{t^2+2t+1}{2}\right) dt + \\ & \int_{0}^{a} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{-t^2+2t+1}{2}\right) dt \\ =& \frac{-c^5+10c^2-15c+6}{120} + \frac{-c^2\left(10a^3-30a^2-30a\right)}{120} - \\ & \frac{\left(-15a^4+20a^3+90a^2+60a\right)c}{120} + \frac{-6a^5+40a^3+60a^2+30a}{120}\end{aligned} \tag{B.6.51}$$

Check: When $a = 0$,

$$\gamma_K(a,c) = \frac{-c^5+10c^2-15c+6}{120} \tag{B.6.52}$$

Check: When $a = c$

$$\gamma_K(a,c) = \frac{-2c^5+10c^4-20c^3+10c^2+15c+6}{120} \tag{B.6.53}$$

d. Case 4(d) $a \in [c, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{0} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{t^2+2t+1}{2}\right)\,dt+\\ & \int_{0}^{c} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{-t^2+2t+1}{2}\right)\,dt+\\ & \int_{c}^{a} \left(\frac{-(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{-t^2+2t+1}{2}\right)\,dt\\ =& \frac{-3c^5+10c^4+20c^3}{120}+\frac{c^2(10a^3-30a^2-30a+10)}{120}+\\ & \frac{c(-15a^4+60a^3-30a^2-60a-15)}{120}+\\ & \frac{6a^5-30a^4+20a^3+60a^2+30a+6}{120}\end{aligned} \tag{B.6.54}$$

Check: When $a = c$,

$$\gamma_K(a,c) = \frac{-2c^5+10c^4-20c^3+10c^2+15c+6}{120} \tag{B.6.55}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{-3c^5+10c^4+20c^3-40c^2-60c+92}{120} \tag{B.6.56}$$

e. Case 4(e) $a \in [1, c+1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{0} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{t^2+2t+1}{2}\right)\,dt+\\ & \int_{0}^{c} \left(\frac{(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{-t^2+2t+1}{2}\right)+\,dt\\ & \int_{c}^{1} \left(\frac{-(t-c)^2+2(t-c)+1}{2}\right)\left(\frac{-t^2+2t+1}{2}\right)\,dt+\\ & \int_{1}^{a} \frac{-(t-c)^2+2(t-c)+1}{2}\,dt\\ =& \frac{-3c^5+10c^4+20c^3+20c^2-8}{120}+\frac{a\left(-60c^2-120c+60\right)}{120}+\\ & \frac{a^2\left(60c+60\right)}{120}-\frac{a^3}{6}\end{aligned} \tag{B.6.57}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{-3c^5+10c^4+20c^3-40c^2-60c+92}{120} \tag{B.6.58}$$

Check: When $a = c + 1$

$$\gamma_K(a,c) = \frac{-3c^5 + 10c^4 - 40c^2 + 60c + 92}{120} \tag{B.6.59}$$

f. Case 4(f) $a \geq c + 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{0} \left(\frac{(t-c)^2 + 2(t-c) + 1}{2}\right)\left(\frac{t^2 + 2t + 1}{2}\right) dt + \\ & \int_{0}^{c} \left(\frac{(t-c)^2 + 2(t-c) + 1}{2}\right)\left(\frac{-t^2 + 2t + 1}{2}\right) dt + \\ & \int_{c}^{1} \left(\frac{-(t-c)^2 + 2(t-c) + 1}{2}\right)\left(\frac{-t^2 + 2t + 1}{2}\right) dt + \\ & \int_{1}^{c+1} \frac{-(t-c)^2 + 2(t-c) + 1}{2}\, dt + \int_{c+1}^{a} 1\, dt \\ =& \frac{-3c^5 + 10c^4 - 40c^2 - 60c + 120a - 28}{120}\end{aligned} \tag{B.6.60}$$

Check: When $a = c + 1$

$$\gamma_K(a,c) = \frac{-3c^5 + 10c^4 - 40c^2 + 60c + 92}{120} \tag{B.6.61}$$

5. **Case 5:** $c \in [1,2]$**:** The intersection of the two CDF is this region is $[0,\infty]$. There are multiple sub-cases to be considered.

a. Case 5(a) $a \leq -1$:

$$\gamma_K(a,c) = 0 \tag{B.6.62}$$

b. Case 5(b) $a \in [c-1, 0]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{a} (1 + 2t - t^2)(1 + 2(t-c) + (t-c)^2)\, dt \\ =& \frac{c^5 - 10c^4 + 20c^3 - 10c^2 - 5c + 4}{120} + \frac{a\,(30c^2 - 60c + 30)}{120} + \\ & \frac{a^2\,(30c^2 - 90c + 60)}{120} + \frac{a^3\,(-10c^2 - 20c + 40)}{120} + \frac{a^4 c}{8} - \frac{a^5}{20}\end{aligned} \tag{B.6.63}$$

Check: When $a = 1$

$$\gamma_K(a,c) = \frac{c^5 - 10c^4 + 20c^3 + 40c^2 - 160c + 128}{120} \tag{B.6.64}$$

c. Case 5(c) $a \in [1, c]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{a}(1+2t-t^2)(1+2(t-c)+(t-c)^2)\,dt + \\ & \frac{1}{2}\int_{-1}^{a}(1+2(t-c)+(t-c)^2)\,dt \\ =& \frac{c^5-10c^4+20c^3-20c^2+20c-60a^2\,(c-1)}{120} + \\ & \frac{60a\,(c-1)^2+20a^3-12}{120}\end{aligned} \tag{B.6.65}$$

check: When $a = c$

$$\gamma_K(a,c) = \frac{c^5-10c^4+40c^3-80c^2+80c-12}{120} \tag{B.6.66}$$

d. Case 5(d) $a \in [c, c+1]$:

$$\begin{aligned}\gamma_K(a,c) =& \frac{1}{4}\int_{-1}^{a}(1+2t-t^2)(1+2(t-c)+(t-c)^2)\,dt + \\ & \frac{1}{2}\int_{-1}^{a}(1+2(t-c)+(t-c)^2)\,dt \\ =& \frac{c^5-10c^4+60c^3}{120} + \frac{(-60a-20)\,c^2}{120} + \\ & \frac{(60a^2-120a+20)\,c-20a^3+60a^2+60a-12}{120}\end{aligned} \tag{B.6.67}$$

Check: When $a = c+1$

$$\gamma_K(a,c) = \frac{c^5-10c^4+40c^3-80c^2+80c+88}{120} \tag{B.6.68}$$

e. Case 5(e) $a \geq c+1$:

$$\begin{aligned}\gamma(a,c) =& \frac{1}{4}\int_{-1}^{a}(1+2t-t^2)(1+2(t-c)+(t-c)^2)\,dt + \\ & \frac{1}{2}\int_{-1}^{a}(1+2(t-c)+(t-c)^2)\,dt + \int_{c+1}^{a}1\,dt \\ =& \frac{c^5-10c^4+40c^3-80c^2-40c+120a-32}{120}\end{aligned} \tag{B.6.69}$$

6. **Case 6:** $c \geq 2$**:**

a. Case 6(a) $a \in [c-1, c]$:

$$\gamma_K(a,c) = \int_{c-1}^{a} \frac{(t-c)^2 + 2(t-c) + 1}{2}\, dt = \frac{(-c+a+1)^3}{6} \tag{B.6.70}$$

Check: When $a = c$

$$\gamma_K(a,c) = \frac{1}{6} \tag{B.6.71}$$

b. Case 6(b) $a \in [c, c+1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{c} \frac{(t-c)^2 + 2(t-c) + 1}{2}\, dt + \int_{c}^{a} \frac{-(t-c)^2 + 2(t-c) + 1}{2}\, dt \\ =& \frac{c^3 + (3-3a)\,c^2 + (3a^2 - 6a - 3)\,c - a^3 + 3a^2 + 3a + 1}{6}\end{aligned} \tag{B.6.72}$$

c. Case 6(c) $a \geq c+1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{c} \frac{(t-c)^2 + 2(t-c) + 1}{2}\, dt + \\ & \int_{c}^{c+1} \frac{-(t-c)^2 + 2(t-c) + 1}{2}\, dt + \int_{c+1}^{a} 1\, dt \\ =& -c + a + \frac{5}{6}\end{aligned} \tag{B.6.73}$$

□

Derivation B.6.5. *Let K be a triangle kernel as in Def B.6.1 with the CDF ν_K. The partial L2-product of the CCDF, $(1-\nu_K)$ at two different central points 0 and $c \in \mathbb{R}$ as in Eqn (5.4.4) is*

1. For $c \geq 2$

$$\xi_K(a,c) = \begin{cases} -a & \text{if } a \leq -1 \\ \frac{a^3+3a^2-3a+1}{6} & \text{if } a \in [0,1] \\ \frac{-a^3+3a^2-3a+1}{6} & \text{if } a \in [-1,0] \\ 0 & \text{if } a \geq 1 \end{cases}$$

2. *For* $c \in [1, 2]$

$$\xi_K(a,c) = \begin{cases} \frac{c^5-10c^4+40c^3-80c^2+80c-120a-32}{120} & \textit{if } a \leq -1 \\ \frac{c^5-10c^4+40c^3-80c^2+80c+20a^3+10a^2-60a}{120} & \textit{if } a \in [-1,0] \\ \frac{c^5-10c^4+40c^3-80c^2+80c-20a^3+60a^2-60a-12}{120} & \textit{if } a \in [0,c-1] \\ \frac{-10c^2+25c+4}{120} - \frac{a^3\left(-10c^2-20c+40\right)}{120} + & \\ \frac{a^2\left(-30c^2+30c+60\right)}{120} - \frac{a\left(-30c^2+60c+30\right)}{120} + \frac{a^4c}{8} - \frac{a^5}{20} & \textit{if } a \in [c-1,1] \\ 0 & \textit{if } a \geq 1 \end{cases}$$

3. *For* $c \in [-1, 0]$

$$\xi_K(a,c) = \begin{cases} \frac{-3c^5+10c^4-40c^2+60c-120a-28}{120} & \textit{if } a \leq -1 \\ \frac{-3c^5+10c^4-40c^2+60c+20a\left(a^2+3a-3\right)-8}{120} & \textit{if } a \in [-1,c-1] \\ \frac{-c^5+10c^2-15c+6}{120} - \frac{c\left(c^4-10c^3+20c^2-30\right)}{120} - \frac{\left(10a^3-30a^2+30a\right)c^2}{120} + & \\ \frac{\left(-15a^4-60a^3-30a^2+60a\right)c}{120} - \frac{6a^5-30a^4-20a^3+60a^2-30a}{120} & \textit{if } a \in [c-1,0] \\ -\frac{c^5+10c^4-20c^3+30c}{120} - \frac{c^5}{120} + \frac{c^2}{12} - \frac{a^3\left(-10c^2-20c+40\right)}{120} + & \\ \frac{-a\left(-30c^2+60c+30\right)}{120} + \frac{a^2\left(-30c^2+30c+60\right)}{120} - \frac{a^4c}{8} - \frac{c}{8} + \frac{a^5}{20} + & \\ \frac{1}{20} & \textit{if } a \in [0,c] \\ -\frac{a\left(30c^2+60c+30\right)}{120} + \frac{a^2\left(30c^2+90c+60\right)}{120} + \frac{10c^2+15c+6}{120} - & \\ \frac{a^3\left(10c^2+60c+60\right)}{120} + \frac{a^4(30+15c)}{120} - \frac{a^5}{20} & \textit{if } a \in [c,1] \\ 0 & \textit{if } a \geq 1 \end{cases}$$

4. *For* $c \in [-1, 0]$

$$\xi_K(a,c) = \begin{cases} \frac{3c^5+10c^4-40c^2-60c+120a-28}{120} & \textit{if } a \le c-1 \\ \frac{3c^5+10c^4-20c^3+20c^2-8}{120} + \frac{-a\left(-60c^2+120c+60\right)}{120} + \frac{a^2(-60c+60)}{120} + \\ \frac{a^3}{6} & \textit{if } a \in [c-1,-1] \\ \frac{c^5}{60} + \frac{c^5-40c^3+30c}{120} + \frac{c^4}{12} + \frac{c^3}{6} + \frac{c^2}{12} - \frac{a\left(-30c^2+60c+30\right)}{120} + \\ \frac{a^2\left(-30c^2+30c+60\right)}{120} - \frac{a^3\left(10c^2-60c+20\right)}{120} - \frac{c}{8} + \frac{a^4(15c-30)}{120} + \\ \frac{-a^5}{20} + \frac{1}{20} & \textit{if } a \in [-1,c] \\ \frac{c^5}{120} + \frac{c^2}{12} + \frac{c}{8} - \frac{a\left(30c^2+60c+30\right)}{120} + \frac{a^2\left(30c^2+90c+60\right)}{120} + \\ \frac{-a^3\left(-10c^2+20c+40\right)}{120} + \frac{a^4c}{8} + \frac{a^5}{20} + \frac{1}{20} & \textit{if } a \in [c,0] \\ \frac{c^5+10c^2+15c+6}{120} - \frac{a\left(30c^2+60c+30\right)}{120} + \frac{a^2\left(30c^2+90c+60\right)}{120} + \\ \frac{-a^3\left(10c^2+60c+60\right)}{120} + \frac{a^4(30+15c)}{120} - \frac{a^5}{20} & \textit{if } a \in [0,c+1] \\ 0 & \textit{if } a \ge c+1 \end{cases}$$

5. *For* $c \in [-2, -1]$

$$\xi_K(a,c) = \begin{cases} \frac{-c^5-10c^4-40c^3-80c^{+}40c-120a-32}{120} & \textit{if } a \le c-1 \\ \frac{-c^5-10c^4-60c^3}{120} - \frac{(60a-20)c^2}{120} + \frac{\left(60a^2+120a+20\right)c}{120} + \\ \frac{20a^3+60a^2-60a-12}{120} & \textit{if } a \in [c-1,c] \\ \frac{-c^5-10c^4-20c^3-20c^2-20c-60a^2(-c-1)-60a(-c-1)^2-20a^3-12}{120} & \textit{if } a \in [c,-1] \\ \frac{-c^5-10c^4-20c^3-10c^2+5c+4}{120} - \frac{a\left(30c^2+60c+30\right)}{120} + \\ \frac{a^2\left(30c^2+90c+60\right)}{120} - \frac{a^3\left(-10c^2+20c+40\right)}{120} - \frac{a^4c}{8} + \frac{a^5}{20} & \textit{if } a \in [-1,c+1] \\ 0 & \textit{if } a \ge c+1 \end{cases}$$

6. *For* $c \ge 2$:

$$\xi_K(a,c) = \begin{cases} c-a & \textit{if } a \le c-1 \\ \frac{-c^3+(3+3a)c^2-\left(3a^2+6a-3\right)c+a^3+3a^2-3a+1}{6} & \textit{if } [c-1,c] \\ \frac{(c-a+1)^3}{120} & \textit{if } a \in [c,c+1] \\ 0 & \textit{if } a \le c+1 \end{cases}$$

Proof. Form Deriv B.6.4, $\int_a^{\infty}(1-\nu_K(t))(1-\nu_K(t-c))\,dt$ is the reflection of $\int_{-\infty}^{a} F(t)F(t-c)\,dt$ on the y-axis.

1. For $c \geq 0$, make a substitution of $c = -c$ and $a = -a$ to $c \leq 0$ of Deriv B.6.4.
2. For $c \leq 0$, make a substitution of $c = -c$ and $a = -a$ to $c \geq 0$ of Deriv B.6.4.

□

B.7 Tricube kernel

Definition B.7.1. *Let the Tricube kernel be defined as*

$$K(u) = \begin{cases} \frac{70}{81}\left(1-|u|^3\right)^3 & \textit{if} \quad |u| \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

The Tricube kernel can be re-written as

$$K(u) = \begin{cases} \frac{70}{81}\left(1+u^3\right)^3 & \textit{if} \quad u \in [-1,0] \\ \frac{70}{81}\left(1-u^3\right)^3 & \textit{if} \quad u \in [0,1] \end{cases}$$

Derivation B.7.1. *Let $K(u)$ be a Tricube kernel as defined in Definition B.7.1. The CDF of the kernel is*

$$\nu_K(a) = \int_{-\infty}^{\infty} K(u)\, du = \begin{cases} 0 & \textit{if} \quad a \leq 0 \\ \frac{81+140a+105a^4+60a^7+14a^{10}}{162} & \textit{if} \quad a \leq 0 \\ \frac{81+140a-105a^4+60a^7-14a^{10}}{162} & \textit{if} \quad a \geq 0 \\ 1 & \textit{if} \quad a \geq 1 \end{cases}$$

Proof. Let the Tricube kernel be defined as in

$$K(u) = \frac{70}{81}(1-|u|^3)^3 \mathbb{1}(u \in [-1,1])$$

There are two cases that we need to consider, $a \in [0,1]$ and $a \in [-1,0]$.

1. **Case 1:** $a \in [-1,0]$

$$\nu_K(a) = \int_{-1}^{t} \frac{70}{81}(1+u^3)^3\, du = \frac{81+140a+105a^4+60a^7+14a^{10}}{162} \tag{B.7.1}$$

2. **Case 2:** $a \in [0,1]$

$$\nu_K(a) = \int_{-1}^{0} \frac{70}{81}(1+u^3)^3\, du + \int_{0}^{a} \frac{70}{81}(1-u^3)^3\, du \tag{B.7.2}$$

$$= \frac{81+140a-105a^4+60a^7-14a^{10}}{162} \tag{B.7.3}$$

□

Derivation B.7.2. *Let $K(u)$ be a Tricube kernel as defined in Def B.7.1. The partial L2-product of the Tricube kernel with two different starting points, 0 and $c \in \mathbb{R}$, $\lambda_K(c)$, is*

$$= \begin{cases} \left(\frac{70}{21}\right)^2 \left(\frac{3c^{19}}{923780} - \frac{3c^{16}}{40040} + \frac{111c^{13}}{20020} - \frac{31c^{10}}{140} + \frac{81c^9}{70} - \frac{729c^8}{220} + \frac{747c^7}{140} - \right. \\ \left. \frac{729c^6}{182} + \frac{9c^4}{5} - \frac{19683c^2}{13090} + \frac{6561}{6916}\right) & \textit{if } c \in [0,1] \\ \left(\frac{70}{81}\right)^2 \left(-\frac{3c^{19}}{923780} - \frac{3c^{16}}{40040} - \frac{111c^{13}}{20020} - \frac{31c^{10}}{140} - \frac{81c^9}{70} - \frac{729c^8}{220} - \frac{747c^7}{140} - \right. \\ \left. \frac{729c^6}{182} + \frac{9c^4}{5} - \frac{19683c^2}{13090} + \frac{6561}{6916}\right) & \textit{if } c \in [-1,0] \\ \left(\frac{70}{81}\right)^2 \left(-\frac{c^{19}}{923780} + \frac{3c^{16}}{40040} - \frac{57c^{13}}{20020} + \frac{31c^{10}}{140} - \frac{81c^9}{70} + \frac{729c^8}{220} - \frac{969c^7}{140} + \right. \\ \left. \frac{9963c^6}{910} - \frac{729c^5}{55} + \frac{66c^4}{5} - \frac{972c^3}{91} + \frac{5832c^2}{935} - \frac{16c}{5} + \frac{2592}{1729}\right) & \textit{if } c \in [1,2] \\ \left(\frac{70}{81}\right)^2 \left(\frac{c^{19}}{923780} + \frac{3c^{16}}{40040} + \frac{57c^{13}}{20020} + \frac{31c^{10}}{140} + \frac{81c^9}{70} - \frac{729c^8}{220} + \frac{969c^7}{140} + \right. \\ \frac{9963c^6}{910} + \frac{729c^5}{55} + \frac{66c^4}{5} + \\ \left. \frac{972c^3}{91} + \frac{5832c^2}{935} + \frac{16c}{5} + \frac{2592}{1729}\right) & \textit{if } c \in [-2,-1] \end{cases}$$

or

$$= \begin{cases} \left(\frac{70}{21}\right)^2 \left(\frac{3|c|^{19}}{923780} - \frac{3|c|^{16}}{40040} + \frac{111|c|^{13}}{20020} - \frac{31|c|^{10}}{140} + \frac{81|c|^9}{70} - \frac{729|c|^8}{220} + \frac{747|c|^7}{140} - \right. \\ \left. \frac{729|c|^6}{182} + \frac{9|c|^4}{5} - \frac{19683|c|^2}{13090} + \frac{6561}{6916}\right) & \textit{if } |c| \in [0,1] \\ \left(\frac{70}{81}\right)^2 \left(-\frac{|c|^{19}}{923780} + \frac{3|c|^{16}}{40040} - \frac{57|c|^{13}}{20020} + \frac{31|c|^{10}}{140} - \frac{81|c|^9}{70} + \frac{729|c|^8}{220} - \frac{969|c|^7}{140} + \right. \\ \left. \frac{9963|c|^6}{910} - \frac{729|c|^5}{55} + \frac{66|c|^4}{5} - \frac{972|c|^3}{91} + \frac{5832|c|^2}{935} - \frac{16|c|}{5} + \frac{2592}{1729}\right) & \textit{if } |c| \in [1,2] \end{cases}$$

Proof. Suppose we have a Tricube kernl as in Def B.7.1. Then, the partial L2-product of the kernel as in Eqn (5.4.3) for Tricube kernel is

$$\lambda_K(c) = \left(\frac{70}{81}\right)^2 \int (1 - |u|^3)^3 (1 - |u|^3)^3 \mathbb{1}(u \in [-1,1]) \mathbb{1}(u \in [-1,1]) \, du \tag{B.7.4}$$

There are two cases to consider: (1) $c \geq 0$; (2) $c \leq 0$.

1. **Case 1:** $c \in [0,1]$**:** There are three cases to be considered under this case.
   (a) Case 1(a) $c \geq 2$:

$$\lambda_K(c) = 0 \tag{B.7.5}$$

(b) Case 1(b) $c \in [0,1]$

$$\lambda_K(c) = \left(\frac{70}{81}\right)^2 \left(\int_{c-1}^{0} (1+u^3)^3(1+(u-c)^3)^3\,du + \int_0^c (1-u^3)^3(1+(u-c)^3)^3\,du + \right.$$
$$\left.\int_c^1 (1-u^3)^3(1-(u-c)^3)^3\,du\right) \tag{B.7.6}$$
$$= \frac{35}{606092058}\big(42c^{19} - 969c^{16} + 71706c^{13} - 2863718c^{10} + 14965236c^9 -$$
$$42854994c^8 + 69006366c^7 - 51802740c^6 + 23279256c^4 - 19446804c^2 + 12269070\big) \tag{B.7.7}$$

(c) Case 1(c) $c \in [1,2]$

$$\lambda_K(c) = \int_{c-1}^{1} \left(\frac{70}{81}\right)^2 (1-u^3)^3(1+(u-c)^3)^3\,du$$
$$= -\left(\frac{70}{81}\right)^2 \left(\frac{14c^{19} - 969c^{16} + 36822c^{13} - 2863718c^{10} + 14965236c^9 -}{12932920}\right.$$
$$\frac{42854994c^8 + 89514282c^7 - 141594156c^6 + 171419976c^5 - 170714544c^4 +}{12932920}$$
$$\left.\frac{138140640c^3 - 80668224c^2 + 41385344c - 19388160}{12932920}\right)$$

2. **Case 2:** $c \leq 0$**:** There are three cases to be considered.

(a) Case 2(a) $c \leq -2$:

$$\lambda_K(c) = 0$$

(b) Case 2(b) $c \in [-1,0]$:

$$\lambda_K(c) = \left(\frac{70}{81}\right)^2 \left(\int_{-1}^{c} (1+u^3)^3(1+(u-c)^3)^3\,du + \int_c^0 (1+u^3)^3(1-(u-c)^3)^3\,du + \right.$$
$$\left.\int_0^{c+1} (1-u^3)^3(1-(u-c)^3)^3\,du\right) \tag{B.7.8}$$
$$= \frac{35}{606092058}\big(-42c^{19} - 969c^{16} - 71706c^{13} - 2863718c^{10} - 14965236c^9 -$$
$$42854994c^8 - 69006366c^7 - 51802740c^6 + 23279256c^4 - 19446804c^2 + 12269070\big) \tag{B.7.9}$$

(c) Case 2(c) $c \in [-2,-1]$:

$$\lambda_K(c) = \int_{-1}^{c+1} \left(\frac{70}{81}\right)^2 (1-(u-c)^3)^3(1+u^3)^3\,du \tag{B.7.10}$$
$$= \frac{35(c+2)^7}{606092058}\big(14c^{12} - 196c^{11} + 1568c^{10} - 8439c^9 + 33474c^8 - 98448c^7 + \tag{B.7.11}$$
$$213558c^6 - 334740c^5 + 561120c^4 - 453722c^3 + 558880c^2 - 206822c + 151470\big) \tag{B.7.12}$$

□

Derivation B.7.3. *Let $K(u)$ be a Tricube kernel as defined in Definition B.7.1. The partial L2-product of two Tricube kernels at two different central points 0 and c where the limit of the integral runs from $-\infty$ to $a \in \mathbb{R}$ is*

Proof. Suppose we have a Tricube kernl as in Def B.7.1. Then, the partial L2-product of the kernel as in Eqn (5.4.3) for Tricube kernel is

$$\lambda_K(c) = \left(\frac{70}{81}\right)^2 \int (1 - |u|^3)^3 (1 - |u|^3)^3 \mathbb{1}(u \in [-1,1]) \mathbb{1}(u \in [-1,1]) \, du \tag{B.7.13}$$

There 4 cases we need to consider: (1) $c \in [0,1]$; (2) $c \in [1,2]$; (3) $c \in [-1,0]$; (4) $c \in [-2,-1]$

1. **Case 1(a):** $c \in [0,1]$**:** The intersection of two kernel functions is in the region $[c-1,1]$. However, the direction of the function changes in between the region.
   (a) Case 1(a:)$a \leq c - 1$:

$$\lambda_K(a,c) = 0 \tag{B.7.14}$$

(b) Case 1(b) $a \in [c-1, 0]$:

$$\lambda_K(a,c) = \int_{c-1}^{a} \frac{70}{81}(1+u^3)^3 \frac{70}{81}(1+(u-c)^3)^3 \, du \tag{B.7.15}$$

$$\begin{aligned}
=&\frac{245c^{19}}{303046029} + \frac{35c^{13}}{34749} + \frac{35c^9}{81} - \frac{245c^8}{198} + \frac{3605c^7}{2187} \\
&- \frac{175c^6}{117} + \frac{2695c^4}{2187} - \frac{105c^2}{187} - \frac{2450c}{6561} + \frac{175}{494} \\
&+ a^{10}\left(-\frac{490c^9}{6561} + \frac{41650c^6}{2187} - \frac{71050c^3}{2187} + \frac{9800}{6561}\right) \\
&+ a^7\left(-\frac{700c^9}{2187} + \frac{20300c^6}{729} - \frac{63700c^3}{2187} + \frac{3500}{2187}\right) \\
&+ a^4\left(-\frac{1225c^9}{2187} + \frac{37975c^6}{2187} - \frac{28175c^3}{2187} + \frac{2450}{2187}\right) \\
&+ a\left(-\frac{4900c^9}{6561} + \frac{4900c^6}{2187} - \frac{4900c^3}{2187} + \frac{4900}{6561}\right) \\
&+ a^5\left(\frac{980c^8}{243} - \frac{19600c^5}{729} + \frac{7840c^2}{729}\right) + a^2\left(\frac{2450c^8}{729} - \frac{4900c^5}{729} + \frac{2450c^2}{729}\right) \\
&+ a^8\left(\frac{1225c^8}{486} - \frac{9800c^5}{243} + \frac{13475c^2}{729}\right) + a^{11}\left(\frac{4900c^8}{8019} - \frac{19600c^5}{729} + \frac{137200c^2}{8019}\right) \\
&+ a^{12}\left(-\frac{4900c^7}{2187} + \frac{57575c^4}{2187} - \frac{12250c}{2187}\right) + a^9\left(-\frac{19600c^7}{2187} + \frac{93100c^4}{2187} - \frac{49000c}{6561}\right) \\
&+ a^3\left(-\frac{19600c^7}{2187} + \frac{24500c^4}{2187} - \frac{4900c}{2187}\right) + a^6\left(-\frac{9800c^7}{729} + \frac{71050c^4}{2187} - \frac{12250c}{2187}\right) \\
&+ a^{13}\left(\frac{137200c^6}{28431} - \frac{39200c^3}{2187} + \frac{24500}{28431}\right) + a^{14}\left(\frac{5950c^2}{729} - \frac{4900c^5}{729}\right) \\
&+ a^{15}\left(\frac{13720c^4}{2187} - \frac{4900c}{2187}\right) + a^{16}\left(\frac{1225}{4374} - \frac{8575c^3}{2187}\right) + \\
&\frac{19600a^{17}c^2}{12393} - \frac{2450a^{18}c}{6561} + \frac{4900a^{19}}{124659}
\end{aligned} \tag{B.7.16}$$

Check: When $a = 0$

$$\begin{aligned}
=&\frac{175}{494} - \frac{2450c}{6561} - \frac{105c^2}{187} + \frac{2695c^4}{2187} - \frac{175c^6}{117} + \frac{3605c^7}{2187} - \\
&\frac{245c^8}{198} + \frac{35c^9}{81} + \frac{35c^{13}}{34749} + \frac{245c^{19}}{303046029}
\end{aligned} \tag{B.7.17}$$

(c) Case 1(c): $a \in [0, c]$

$$\begin{aligned}
\lambda_K(a,c) = & \left(\frac{70}{81}\right)^2 \int_{c-1}^{0} (1+u^3)^3(1+(u-c)^3)^3 \, du + \\
& \left(\frac{70}{81}\right)^2 \int_{0}^{a} (1-u^3)^3(1+(u-c)^3)^3 \, du \qquad \text{(B.7.18)}
\end{aligned}$$

$$\begin{aligned}
= & \frac{245c^{19}}{303046029} + \frac{35c^{13}}{34749} + \frac{35c^9}{81} + \frac{3605c^7}{2187} - \frac{175c^6}{117} - \frac{105c^2}{187} - \frac{2450c}{1656} + \frac{175}{494} \\
& + a^4\left(\frac{1225c^9}{2187} + \frac{30625c^6}{2187} - \frac{20825c^3}{2187}\right) + a^{10}\left(\frac{490c^9}{6561} + \frac{40670c^6}{2187} + \frac{12250c^3}{2187}\right) + \\
& a^7\left(-\frac{700c^9}{2187} - \frac{700c^6}{27} + \frac{20300c^3}{2187} - \frac{700}{2187}\right) + \\
& a\left(-\frac{4900c^9}{6561} + \frac{4900c^6}{2187} - \frac{4900c^3}{2187} + \frac{4900}{6561}\right) + a^2\left(\frac{2450c^8}{729} - \frac{4900c^5}{729} + \frac{2450c^2}{729}\right) + \\
& a^8\left(\frac{1225c^8}{486} + \frac{2450c^5}{81} - \frac{4900c^2}{729}\right) - \frac{245c^8}{198} + \\
& a^{11}\left(-\frac{4900c^8}{8019} - \frac{196000c^5}{8019} + \frac{9800c^2}{8019}\right) + a^5\left(-\frac{980c^8}{243} - \frac{7840c^5}{729} + \frac{1960c^2}{729}\right) + \\
& a^6\left(\frac{9800c^7}{729} - \frac{2450c^4}{2187} + \frac{2450c}{2187}\right) + a^{12}\left(\frac{4900c^7}{2187} + \frac{45325c^4}{2187} - \frac{2450c}{2187}\right) + \\
& a^3\left(-\frac{19600c^7}{2187} + \frac{24500c^4}{2187} - \frac{4900c}{2187}\right) a^9\left(-\frac{19600c^7}{2187} - \frac{4900c^4}{243} + \frac{9800c}{6561}\right) + \\
& a^{13}\left(-\frac{137200c^6}{28431} - \frac{313600c^3}{28431} + \frac{4900}{28431}\right) + a^{14}\left(\frac{4900c^5}{729} + \frac{2450c^2}{729}\right) + \frac{2695c^4}{2187} + \\
& a^{15}\left(-\frac{13720c^4}{2187} - \frac{980c}{2187}\right) + \frac{8575a^{16}c^3}{2187} - \frac{19600a^{17}c^2}{12393} + \frac{2450a^{18}c}{6561} - \frac{4900a^{19}}{124659}
\end{aligned}$$

(B.7.19)

Check: When $a = c$

$$\begin{aligned}
= & \frac{175}{494} + \frac{2450c}{6561} - \frac{105c^2}{187} + \frac{245c^4}{2187} - \frac{175c^6}{117} + \frac{5110c^7}{2187} - \frac{245c^8}{198} + \frac{35c^9}{81} - \\
& \frac{1085c^{10}}{6561} + \frac{980c^{13}}{312741} - \frac{35c^{16}}{625482} + \frac{490c^{19}}{303046029}
\end{aligned}$$

(d) Case 1(d): $a \in [c, 1]$

$$= \left(\frac{70}{81}\right)^2 \int_{c-1}^{0} (1+u^3)^3(1+(u-c)^3)^3\,du + \left(\frac{70}{81}\right)^2 \int_{0}^{c} (1-u^3)^3(1+(u-c)^3)^3\,du + \left(\frac{70}{81}\right)^2 \int_{c}^{a} (1-u^3)^3(1-(u-c)^3)^3\,du \tag{B.7.20}$$

$$= \frac{245c^{19}}{101015343} - \frac{35c^{16}}{625482} + \frac{1295c^{13}}{312741} - \frac{1085c^{10}}{6561} + \frac{35c^9}{81} - \frac{245c^8}{198} + \frac{3815c^7}{2187} - \frac{175c^6}{117} + \frac{245c^4}{2187} - \frac{105c^2}{187} - \frac{2450c}{6561} + \frac{175}{494} + \tag{B.7.21}$$

$$a\left(\frac{4900c^9}{6561} + \frac{4900c^6}{2187} + \frac{4900c^3}{2187} + \frac{4900}{6561}\right) + a^2\left(-\frac{2450c^8}{729} - \frac{4900c^5}{729} - \frac{2450c^2}{729}\right) + a^3\left(\frac{19600c^7}{2187} + \frac{24500c^4}{2187} + \frac{4900c}{2187}\right) + a^4\left(-\frac{1225c^9}{2187} - \frac{37975c^6}{2187} - \frac{28175c^3}{2187} - \frac{2450}{2187}\right) + a^5\left(\frac{980c^8}{243} + \frac{19600c^5}{729} + \frac{7840c^2}{729}\right) + a^6\left(-\frac{9800c^7}{729} - \frac{71050c^4}{2187} - \frac{12250c}{2187}\right) + a^7\left(\frac{700c^9}{2187} + \frac{20300c^6}{729} + \frac{63700c^3}{2187} + \frac{3500}{2187}\right) + a^8\left(-\frac{1225c^8}{486} - \frac{9800c^5}{243} - \frac{13475c^2}{729}\right) + a^9\left(\frac{19600c^7}{2187} + \frac{93100c^4}{2187} + \frac{49000c}{6561}\right) + a^{10}\left(-\frac{490c^9}{6561} - \frac{41650c^6}{2187} - \frac{71050c^3}{2187} - \frac{9800}{6561}\right) + \tag{B.7.22}$$

$$a^{11}\left(\frac{4900c^8}{8019} + \frac{19600c^5}{729} + \frac{137200c^2}{8019}\right) + a^{12}\left(-\frac{4900c^7}{2187} - \frac{57575c^4}{2187} - \frac{12250c}{2187}\right) + a^{13}\left(\frac{137200c^6}{28431} + \frac{39200c^3}{2187} + \frac{24500}{28431}\right) + a^{14}\left(-\frac{4900c^5}{729} - \frac{5950c^2}{729}\right) + a^{15}\left(\frac{13720c^4}{2187} + \frac{4900c}{2187}\right) + a^{16}\left(-\frac{8575c^3}{2187} - \frac{1225}{4374}\right) + \frac{19600a^{17}c^2}{12393} - \frac{2450a^{18}c}{6561} + \frac{4900a^{19}}{124659} \tag{B.7.23}$$

Check: When a = 1

$$\frac{175}{247} - \frac{210c^2}{187} + \frac{980c^4}{729} - \frac{350c^6}{117} + \frac{2905c^7}{729} - \frac{245c^8}{99} + \frac{70c^9}{81} - \frac{1085c^{10}}{6561} \tag{B.7.24}$$

$$+ \frac{1295c^{13}}{312741} - \frac{35c^{16}}{625482} + \frac{245c^{19}}{101015343} \tag{B.7.25}$$

2. **Case 2** $c \in [1, 2]$

(a) Case 2(a) $a \leq c - 1$:

$$\lambda_K(a, c) = 0 \tag{B.7.26}$$

(b) Case 2(b) $a \in [c-1, a]$:

$$\lambda_K(a,c) = \left(\frac{70}{81}\right)^2 \int_{c-1}^{a} (1-u^3)^3(1+(u-c)^3)^3\, du \tag{B.7.27}$$

$$\begin{aligned} &= -\frac{245c^{19}}{303046029} + \frac{35c^{16}}{625482} - \frac{665c^{13}}{312741} + \frac{1085c^{10}}{6561} - \frac{35c^9}{81} + \\ &\frac{245c^8}{198} - \frac{2135c^7}{729} + \frac{1435c^6}{351} - \frac{490c^5}{99} + \frac{12005c^4}{2187} - \frac{1400c^3}{351} + \\ &\frac{3920c^2}{1683} - \frac{3430c}{2187} + \frac{11200}{20007} + a\left(-\frac{4900c^9}{6561} + \frac{4900c^6}{2187} - \frac{4900c^3}{2187} + \frac{4900}{6561}\right) + \\ &a^2\left(\frac{2450c^8}{729} - \frac{4900c^5}{729} + \frac{2450c^2}{729}\right) + a^3\left(-\frac{19600c^7}{2187} + \frac{24500c^4}{2187} - \frac{4900c}{2187}\right) + \\ &a^4\left(\frac{1225c^9}{2187} + \frac{30625c^6}{2187} - \frac{20825c^3}{2187}\right) + a^5\left(-\frac{980c^8}{243} - \frac{7840c^5}{729} + \frac{1960c^2}{729}\right) + \\ &a^6\left(\frac{9800c^7}{729} - \frac{2450c^4}{2187} + \frac{2450c}{2187}\right) + a^7\left(-\frac{700c^9}{2187} - \frac{700c^6}{27} + \frac{20300c^3}{2187} - \frac{700}{2187}\right) + \\ &a^8\left(\frac{1225c^8}{486} + \frac{2450c^5}{81} - \frac{4900c^2}{729}\right) + a^9\left(-\frac{19600c^7}{2187} - \frac{4900c^4}{243} + \frac{9800c}{6561}\right) + \\ &a^{10}\left(\frac{490c^9}{6561} + \frac{40670c^6}{2187} + \frac{12250c^3}{2187}\right) + a^{11}\left(-\frac{4900c^8}{8019} - \frac{196000c^5}{8019} + \frac{9800c^2}{8019}\right) + \\ &a^{12}\left(\frac{4900c^7}{2187} + \frac{45325c^4}{2187} - \frac{2450c}{2187}\right) + a^{13}\left(-\frac{137200c^6}{28431} - \frac{313600c^3}{28431} + \frac{4900}{28431}\right) + \\ &a^{14}\left(\frac{4900c^5}{729} + \frac{2450c^2}{729}\right) + a^{15}\left(-\frac{13720c^4}{2187} - \frac{980c}{2187}\right) + \frac{8575a^{16}c^3}{2187} - \\ &\frac{19600a^{17}c^2}{12393} + \frac{2450a^{18}c}{6561} - \frac{4900a^{19}}{124659} \end{aligned} \tag{B.7.28}$$

Check:When $a = 1$

$$\begin{aligned} =&\frac{22400}{20007} - \frac{15680c}{6561} + \frac{7840c^2}{1683} - \frac{2800c^3}{351} + \frac{21560c^4}{2187} - \\ &\frac{98\theta c^5}{99} + \frac{287\theta c^6}{351} - \frac{11305c^7}{2187} + \frac{245c^8}{99} - \frac{70c^9}{81} + \frac{1085c^{10}}{6561} - \\ &\frac{665c^{13}}{312741} + \frac{35c^{16}}{625482} - \frac{245c^{19}}{303046629} \end{aligned} \tag{B.7.29}$$

3. **Case 3:** $c \in [-1, 0]$**:** When $c \in [-1, 0]$, the intersection region of the two kernels is $[-1, c+1]$

(a) Case 3(a): $a \leq -1$:

$$\lambda_K(a,c) = 0 \tag{B.7.30}$$

(b) Case 3(b): $a \in [-1, c]$

$$\lambda_K(a,c) = \left(\frac{70}{81}\right)^2 \int_{-1}^{a} (1+u^3)^3(1+(u-c)^3)^3 \, du \tag{B.7.31}$$

$$\begin{aligned}
&= -\frac{35c^9}{81} - \frac{245c^8}{198} - \frac{4900c^7}{2187} - \frac{175c^6}{117} + \frac{2695c^4}{2187} - \frac{105c^2}{187} - \frac{2450c}{6561} + \frac{175}{494} + \\
&a\left(-\frac{4900c^9}{6561} + \frac{4900c^6}{2187} - \frac{4900c^3}{2187} + \frac{4900}{6561}\right) + \\
&a^2\left(\frac{2450c^8}{729} - \frac{4900c^5}{729} + \frac{2450c^2}{729}\right) + a^3\left(-\frac{19600c^7}{2187} + \frac{24500c^4}{2187} - \frac{4900c}{2187}\right) + \\
&a^4\left(-\frac{1225c^9}{2187} + \frac{37975c^6}{2187} - \frac{28175c^3}{2187} + \frac{2450}{2187}\right) + \\
&a^5\left(\frac{980c^8}{243} - \frac{19600c^5}{729} + \frac{7840c^2}{729}\right) + a^6\left(-\frac{9800c^7}{729} + \frac{71050c^4}{2187} - \frac{12250c}{2187}\right) + \\
&a^7\left(-\frac{700c^9}{2187} + \frac{20300c^6}{729} - \frac{63700c^3}{2187} + \frac{3500}{2187}\right) + \\
&a^8\left(\frac{1225c^8}{486} - \frac{9800c^5}{243} + \frac{13475c^2}{729}\right) + a^9\left(-\frac{19600c^7}{2187} + \frac{93100c^4}{2187} - \frac{49000c}{6561}\right) + \\
&a^{10}\left(-\frac{490c^9}{6561} + \frac{41650c^6}{2187} - \frac{71050c^3}{2187} + \frac{9800}{6561}\right) + \\
&a^{11}\left(\frac{4900c^8}{8019} - \frac{19600c^5}{729} + \frac{137200c^2}{8019}\right) + \\
&a^{12}\left(-\frac{4900c^7}{2187} + \frac{57575c^4}{2187} - \frac{12250c}{2187}\right) + a^{13}\left(\frac{137200c^6}{28431} - \frac{39200c^3}{2187} + \frac{24500}{28431}\right) + \\
&a^{14}\left(\frac{5950c^2}{729} - \frac{4900c^5}{729}\right) + a^{15}\left(\frac{13720c^4}{2187} - \frac{4900c}{2187}\right) + a^{16}\left(\frac{1225}{4374} - \frac{8575c^3}{2187}\right) + \\
&\frac{19600a^{17}c^2}{12393} - \frac{2450a^{18}c}{6561} + \frac{4900a^{19}}{124659}
\end{aligned} \tag{B.7.32}$$

Check:When $a = c$,

$$\begin{aligned}
=&\frac{175}{494} + \frac{2450c}{6561} - \frac{105c^2}{187} + \frac{2695c^4}{2187} - \frac{175c^6}{117} - \frac{3605c^7}{2187} - \frac{245c^8}{198} - \frac{35c^9}{81} - \frac{35c^{13}}{34749} - \\
&\frac{245c^{19}}{303046029}
\end{aligned} \tag{B.7.33}$$

(c) Case 3(c) $a \in [c, 0]$

$$\begin{aligned}\lambda_K(a,c) = &\left(\frac{70}{81}\right)^2 \int_{-1}^{c} (1+u^3)^3(1+(u-c)^3)^3 \, du + \\ &\left(\frac{70}{81}\right)^2 \int_{c}^{a} (1+u^3)^3(1-(u-c)^3)^3 \, du\end{aligned} \tag{B.7.34}$$

$$\begin{aligned} = & -\frac{490c^{19}}{303046029} - \frac{35c^{16}}{625482} - \frac{980c^{13}}{312741} - \frac{1085c^{10}}{6561} - \frac{35c^9}{81} - \frac{245c^8}{198} \\ & - \frac{5110c^7}{2187} - \frac{175c^6}{117} + \frac{245c^4}{2187} - \frac{105c^2}{187} - \frac{2450c}{6561} + \frac{175}{494} + \\ & a\left(\frac{4900c^9}{6561} + \frac{4900c^6}{2187} + \frac{4900c^3}{2187} + \frac{4900}{6561}\right) + \\ & a^2\left(-\frac{2450c^8}{729} - \frac{4900c^5}{729} - \frac{2450c^2}{729}\right) + a^3\left(\frac{19600c^7}{2187} + \frac{24500c^4}{2187} + \frac{4900c}{2187}\right) + \\ & a^4\left(\frac{1225c^9}{2187} - \frac{30625c^6}{2187} - \frac{20825c^3}{2187}\right) + a^5\left(-\frac{980c^8}{243} + \frac{7840c^5}{729} + \frac{1960c^2}{729}\right) + \\ & a^6\left(\frac{9800c^7}{729} + \frac{2450c^4}{2187} + \frac{2450c}{2187}\right) + a^7\left(\frac{700c^9}{2187} - \frac{700c^6}{27} - \frac{20300c^3}{2187} - \frac{700}{2187}\right) + \\ & a^8\left(-\frac{1225c^8}{486} + \frac{2450c^5}{81} + \frac{4900c^2}{729}\right) + a^9\left(\frac{19600c^7}{2187} - \frac{4900c^4}{243} - \frac{9800c}{6561}\right) + \\ & a^{10}\left(\frac{490c^9}{6561} - \frac{40670c^6}{2187} + \frac{12250c^3}{2187}\right) + a^{11}\left(-\frac{4900c^8}{8019} + \frac{196000c^5}{8019} + \frac{9800c^2}{8019}\right) + \\ & a^{12}\left(\frac{4900c^7}{2187} - \frac{45325c^4}{2187} - \frac{2450c}{2187}\right) + a^{13}\left(-\frac{137200c^6}{28431} + \frac{313600c^3}{28431} + \frac{4900}{28431}\right) + \\ & a^{14}\left(\frac{4900c^5}{729} - \frac{2450c^2}{729}\right) + a^{15}\left(\frac{980c}{2187} - \frac{13720c^4}{2187}\right) + \\ & \frac{8575a^{16}c^3}{2187} - \frac{19600a^{17}c^2}{12393} + \frac{2450a^{18}c}{6561} - \frac{4900a^{19}}{124659}\end{aligned} \tag{B.7.35}$$

Check: When $a = 0$

$$\begin{aligned} = & \frac{175}{494} - \frac{2450c}{6561} - \frac{105c^2}{187} + \frac{245c^4}{2187} - \frac{175c^6}{117} - \frac{5110c^7}{2187} - \frac{245c^8}{198} - \frac{35c^9}{81} - \frac{1085c^{10}}{6561} - \\ & \frac{980c^{13}}{312741} - \frac{35c^{16}}{625482} - \frac{490c^{19}}{303046029}\end{aligned} \tag{B.7.36}$$

(d) Case 3(d) $a \in [0, c+1]$:

$$\begin{aligned}
\lambda_K(a,c) = &\left(\frac{70}{81}\right)^2 \int_{-1}^{c} (1+u^3)^3(1+(u-c)^3)^3\, du + \\
&\left(\frac{70}{81}\right)^2 \int_{c}^{0} (1+u^3)^3(1-(u-c)^3)^3\, du + \\
&\left(\frac{70}{81}\right)^2 \int_{0}^{a} (1-u^3)^3(1-(u-c)^3)^3\, du
\end{aligned} \tag{B.7.37}$$

$$\begin{aligned}
= &-\frac{490c^{19}}{303046029} - \frac{35c^{16}}{625482} - \frac{980c^{13}}{312741} - \frac{1085c^{10}}{6561} - \frac{35c^9}{81} \\
&- \frac{245c^8}{198} - \frac{5110c^7}{2187} - \frac{175c^6}{117} + \frac{245c^4}{2187} - \frac{105c^2}{187} - \frac{2450c}{6561} + \frac{175}{494} \\
&+ a\left(\frac{4900c^9}{6561} + \frac{4900c^6}{2187} + \frac{4900c^3}{2187} + \frac{4900}{6561}\right) \\
&+ a^2\left(-\frac{2450c^8}{729} - \frac{4900c^5}{729} - \frac{2450c^2}{729}\right) + a^3\left(\frac{19600c^7}{2187} + \frac{24500c^4}{2187} + \frac{4900c}{2187}\right) \\
&+ a^4\left(-\frac{1225c^9}{2187} - \frac{37975c^6}{2187} - \frac{28175c^3}{2187} - \frac{2450}{2187}\right) \\
&+ a^5\left(\frac{980c^8}{243} + \frac{19600c^5}{729} + \frac{7840c^2}{729}\right) + a^6\left(-\frac{9800c^7}{729} - \frac{71050c^4}{2187} - \frac{12250c}{2187}\right) \\
&+ a^7\left(\frac{700c^9}{2187} + \frac{20300c^6}{729} + \frac{63700c^3}{2187} + \frac{3500}{2187}\right) \\
&+ a^8\left(-\frac{1225c^8}{486} - \frac{9800c^5}{243} - \frac{13475c^2}{729}\right) + a^9\left(\frac{19600c^7}{2187} + \frac{93100c^4}{2187} + \frac{49000c}{6561}\right) \\
&+ a^{10}\left(-\frac{490c^9}{6561} - \frac{41650c^6}{2187} - \frac{71050c^3}{2187} - \frac{9800}{6561}\right) \\
&+ a^{11}\left(\frac{4900c^8}{8019} + \frac{19600c^5}{729} + \frac{137200c^2}{8019}\right) + a^{12}\left(-\frac{4900c^7}{2187} - \frac{57575c^4}{2187} - \frac{12250c}{2187}\right) \\
&+ a^{13}\left(\frac{137200c^6}{28431} + \frac{39200c^3}{2187} + \frac{24500}{28431}\right) + a^{14}\left(-\frac{4900c^5}{729} - \frac{5950c^2}{729}\right) \\
&+ a^{15}\left(\frac{13720c^4}{2187} + \frac{4900c}{2187}\right) + a^{16}\left(-\frac{8575c^3}{2187} - \frac{1225}{4374}\right) \\
&+ \frac{19600a^{17}c^2}{12393} - \frac{2450a^{18}c}{6561} + \frac{4900a^{19}}{124659}
\end{aligned} \tag{B.7.38}$$

Check: When $a = c+1$

$$= \frac{175}{247} - \frac{210c^2}{187} + \frac{980c^4}{729} - \frac{350c^6}{117} - \frac{2905c^7}{729} - \frac{245c^8}{99} - \frac{70c^9}{81} - \frac{1085c^{12}}{6561} - \tag{B.7.39}$$

$$\frac{1295c^{13}}{312741} - \frac{35c^{16}}{625482} - \frac{245c^{19}}{101015343} \tag{B.7.40}$$

4. **Case 4:** $c \in [-2, -1]$

(a) Case 4(a) $a \leq -1$

$$\lambda_K(a,c) = 0 \tag{B.7.41}$$

(b) Case 4(b): $c \in [-2, -1]$

$$\lambda_K(a,c) = \left(\frac{70}{81}\right)^2 \int_{-1}^{a} (1+u^3)^3(1-(u-c)^3)^3 \, du \tag{B.7.42}$$

$$\begin{aligned}
&= \frac{35c^9}{81} + \frac{245c^8}{198} + \frac{4900c^7}{2187} + \frac{1435c^6}{351} + \frac{490c^5}{99} \\
&+ \frac{3185c^4}{729} + \frac{1400c^3}{351} + \frac{3920c^2}{1683} + \frac{5390c}{6561} + \frac{11200}{20007} \\
&+ a\left(\frac{4900c^9}{6561} + \frac{4900c^6}{2187} + \frac{4900c^3}{2187} + \frac{4900}{6561}\right) \\
&+ a^2\left(-\frac{2450c^8}{729} - \frac{4900c^5}{729} - \frac{2450c^2}{729}\right) + a^3\left(\frac{19600c^7}{2187} + \frac{24500c^4}{2187} + \frac{4900c}{2187}\right) \\
&+ a^4\left(\frac{1225c^9}{2187} - \frac{30625c^6}{2187} - \frac{20825c^3}{2187}\right) + a^5\left(-\frac{980c^8}{243} + \frac{7840c^5}{729} + \frac{1960c^2}{729}\right) \\
&+ a^6\left(\frac{9800c^7}{729} + \frac{2450c^4}{2187} + \frac{2450c}{2187}\right) + a^7\left(\frac{700c^9}{2187} - \frac{700c^6}{27} - \frac{20300c^3}{2187} - \frac{700}{2187}\right) \\
&+ a^8\left(-\frac{1225c^8}{486} + \frac{2450c^5}{81} + \frac{4900c^2}{729}\right) + a^9\left(\frac{19600c^7}{2187} - \frac{4900c^4}{243} - \frac{9800c}{6561}\right) \\
&+ a^{10}\left(\frac{490c^9}{6561} - \frac{40670c^6}{2187} + \frac{12250c^3}{2187}\right) + a^{11}\left(-\frac{4900c^8}{8019} + \frac{196000c^5}{8019} + \frac{9800c^2}{8019}\right) \\
&+ a^{12}\left(\frac{4900c^7}{2187} - \frac{45325c^4}{2187} - \frac{2450c}{2187}\right) + a^{13}\left(-\frac{137200c^6}{28431} + \frac{313600c^3}{28431} + \frac{4900}{28431}\right) \\
&+ a^{14}\left(\frac{4900c^5}{729} - \frac{2450c^2}{729}\right) \\
&+ a^{15}\left(\frac{980c}{2187} - \frac{13720c^4}{2187}\right) + \frac{8575a^{16}c^3}{2187} - \frac{19600a^{17}c^2}{12393} + \frac{2450a^{18}c}{6561} - \frac{4900a^{19}}{124659}
\end{aligned} \tag{B.7.43}$$

Check:When $a = c + 1$

$$\begin{aligned}
&= \frac{22400}{20007} + \frac{15680c}{6561} + \frac{7840c^2}{1683} + \frac{2800c^3}{351} + \frac{21560c^4}{2187} + \frac{980c^5}{99} + \frac{2870c^6}{351} + \\
&\frac{11305c^7}{2187} + \frac{245c^8}{99} + \frac{70c^9}{81} + \frac{1085c^{18}}{6561} + \frac{665c^{13}}{312741} + \frac{35c^{16}}{625482} + \frac{245c^{19}}{303046029}
\end{aligned} \tag{B.7.44}$$

□

Derivation B.7.4. *Let K be a Tricube kernel as in Def B.7.1 and let its CDF as in Deriv B.7.1. The partial L2-product of the Tricube kernel at two different centre points 0 and $c \in \mathbb{R}$ for the integral boundary $(-\infty, a]$ for $a \in \mathbb{R}$ are show below.*

Proof. Suppose K is a Tricube kernel. The partial L2-product of the Tricube CDF is

$$\begin{aligned}
\gamma_K(a,c) = \int_{-\infty}^{a} & \left(14t^{10} + 60t^7 + 105t^4 + 140t + 81\right) \mathbb{1}(t \in [-1,1]) \times \\
& \left(14\,(t-c)^{10} + 60\,(t-c)^7 + 105\,(t-c)^4 + 140\,(t-c) + 81\right) \mathbb{1}(t \in [c-1, c+1]) \, dt
\end{aligned}$$

1. **Case 1:** $c \leq -2$**:**
a. Case 1(a) $a \leq -1$:

$$\gamma_{1K}(a,c) = 0 \tag{B.7.45}$$

b. Case 1(b) $a \in [-1, 0]$:

$$\gamma_{2K}(a,c) = \int_{-1}^{a} \frac{\left(14t^{10} + 60t^7 + 105t^4 + 140t + 81\right)}{162}\, dt = \frac{7a^{11}}{891} + \frac{5a^8}{108} + \frac{7a^5}{54} + \frac{35a^2}{81} + \frac{a}{2} + \frac{7}{44} \tag{B.7.46}$$

c. Case 1(c) $a \in [0, 1]$:

$$\begin{aligned}\gamma_{3K}(a,c) =& \int_{-1}^{0} \frac{\left(14t^{10} + 60t^7 + 105t^4 + 140t + 81\right)}{162}\, dt + \int_{0}^{a} \frac{\left(-14t^{10} + 60t^7 - 105t^4 + 140t + 81\right)}{162}\, dt \\ =& \frac{7}{44} - \left(\frac{7a^{11}}{891} + \frac{5a^8}{108} - \frac{7a^5}{54} + \frac{35a^2}{81} + \frac{a}{2}\right)\end{aligned} \tag{B.7.47}$$

d. Case 1(d) $a \geq 1$:

$$\begin{aligned}\gamma_{4K}(a,c) =& \int_{-1}^{0} \frac{\left(14t^{10} + 60t^7 + 105t^4 + 140t + 81\right)}{162}\, dt + \\ & \int_{0}^{1} \frac{\left(-14t^{10} + 60t^7 - 105t^4 + 140t + 81\right)}{162}\, dt + \int_{1}^{a} 1\, dt \\ =& a\end{aligned} \tag{B.7.48}$$

2. **Case 2:** $c \in [-2, -1]$**:** The intersection of two CDF in this region is $[-1, \infty]$. The cases are below.
a. Case 2(a) $a \leq -1$:

$$\gamma_{5K}(a,c) = 0 \tag{B.7.49}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}
\gamma_{6K}(a,c) = & \int_{-1}^{a} \frac{\left(14t^{10} + 60t^{7} + 105t^{4} + 140t + 81\right)}{162} \times \\
& \frac{\left(-14\,(t-c)^{10} + 60\,(t-c)^{7} - 105\,(t-c)^{4} + 140\,(t-c) + 81\right)}{162}\, dt \\
= & -\frac{\left(4938024a^{11} + 29099070a^{8} + 81477396a^{5} + 201753552a^{2} + 314269956a + 169832754\right) c^{10}}{277134 \times 162^{2}} - \\
& \frac{\left(-45265220a^{12} - 258658400a^{9} - 678978300a^{6} - 1345023680a^{3} - 1571349780a^{2} + 691911220\right) c^{9}}{277134 \times 162^{2}} - \\
& \frac{\left(188024760a^{13} + 1047566520a^{10} + 2618916300a^{7} + 4539454920a^{4} + 4714049340a^{3} + 1933968960\right) c^{8}}{277134 \times 162^{2}} - \\
& \frac{\left(-465585120a^{14} - 2518392240a^{11} - 5986094400a^{8} - 9334981656a^{5} - 9428098680a^{4} + 864658080a^{2}\right) c}{277134 \times 162^{2}} + \\
& \frac{\left(1346871240a + 4508617464\right) c^{7}}{277134 \times 162^{2}} - \\
& \frac{\left(760455696a^{15} + 3938074140a^{12} + 8729721000a^{9} + 12085813740a^{6} + 13199338152a^{5} - 4035071040a^{3}\right) c}{277134 \times 162^{2}} - \\
& \frac{\left(4714049340a^{2} + 7344605268\right) c^{6}}{277134 \times 162^{2}} - \\
& \frac{\left(-855512658a^{16} - 4136544720a^{13} - 8171018856a^{10} - 9288423144a^{7} - 13199338152a^{6} + 9078909840a^{4}\right)}{277134 \times 162^{2}} + \\
& \frac{\left(9428098680a^{3} + 9150090642\right) c^{5}}{277134 \times 162^{2}} - \\
& \frac{\left(670990320a^{17} + 2909907000a^{14} + 4639978980a^{11} + 3171798630a^{8} + 9428098680a^{7} - 11494132650a^{5}\right) c^{4}}{277134 \times 162^{2}} - \\
& \frac{\left(11785123350a^{4} + 1513151640a^{2} + 2357024670a + 9792226080\right) c^{4}}{277134 \times 162^{2}} - \\
& \frac{\left(-362121760a^{18} - 1319157840a^{15} - 1299758460a^{12} + 633713080a^{9} - 4714049340a^{8} + 8050742700a^{6}\right) c^{3}}{277134 \times 162^{2}} + \\
& \frac{\left(9428098680a^{5} - 4035071040a^{3} - 4714049340a^{2} + 7746819080\right) c^{3}}{277134 \times 162^{2}} - \\
& \frac{\left(128648520a^{19} + 349188840a^{16} - 13430340a^{13} - 803134332a^{10} + 1571349780a^{9} - 2569032180a^{7}\right) c^{2}}{277134 \times 162^{2}} - \\
& \frac{\left(4714049340a^{6} + 4539454920a^{4} + 4714049340a^{3} + 4460125032\right) c^{2}}{277134 \times 162^{2}} - \\
& \frac{\left(-27159132a^{20} - 41081040a^{17} + 91454220a^{14} + 146024424a^{11} - 314269956a^{10} + 201614985a^{8}\right) c}{277134 \times 162^{2}} + \\
& \frac{\left(1346871240a^{7} - 1815781968a^{5} - 2357024670a^{4} + 1498740672a^{2} + 2334576816a + 2877253353\right) c}{277134 \times 162^{2}} - \\
& \frac{2586584a^{21} - 12193896a^{15} - 44803330a^{9} - 336717810a^{8} - 999160448a^{3} - 2334576816a^{2} - 1818276174a}{277134 \times 162^{2}} - \\
& \frac{200552638}{277134 \times 162^{2}}
\end{aligned} \tag{B.7.50}$$

Check: When $a = c + 1$,

$$\begin{aligned}
\gamma_{6K}(a = c+1, c) = & \frac{-14c^{21} - 1330c^{18} - 73644c^{15} - 6995534c^{12} + 288874404c^{10} + 1705990715c^{9}}{277134 \times 162^{2}} + \\
& \frac{5560160760c^{8} + 12532704912c^{7} + 20935810896c^{6} + 26954692380c^{5} + 27556041240c^{4}}{277134 \times 162^{2}} + \\
& \frac{21719176336c^{3} + 13078646856c^{2} + 8769254832c + 5743694528}{277134 \times 162^{2}}
\end{aligned} \tag{B.7.51}$$

c. Case 2(c) $a \in [c+1, 0]$:

$$\begin{aligned}\gamma_{7K}(a,c) =&\gamma_{6K}(a=c+1,c) + \int_{c+1}^{a} \frac{\left(14t^{10}+60t^{7}+105t^{4}+140t+81\right)}{162}\,dt \\ =&\gamma_{6K}(a=c+1,c) + \frac{28\left(a^{11}-(c+1)^{11}\right)+165\left(a^{8}-(c+1)^{8}\right)+462\left(a^{5}-(c+1)^{5}\right)+1540a^{2}}{3564} + \\ &\frac{1782a-1540c^{2}-4862c-3322}{3564}\end{aligned} \tag{B.7.52}$$

Check: When $a = 0$

$$\begin{aligned}\gamma_{7K}(a=0,c+1) =&\gamma_{6K}(a=c+1,c) + \int_{c+1}^{0} \frac{\left(14t^{10}+60t^{7}+105t^{4}+140t+81\right)}{162}\,dt \\ =&\gamma_{6K}(a=c+1,c) - \frac{28c^{11}+308c^{10}+1540c^{9}+4785c^{8}+10560c^{7}+17556c^{6}+22638c^{5}}{22\times 162} - \\ &\frac{23100c^{4}+18480c^{3}+11924c^{2}+8008c+3581}{22\times 162}\end{aligned} \tag{B.7.53}$$

d. Case 2(d) $a \in [0, 1]$:

$$\begin{aligned}\gamma_{8K}(a,c) =&\gamma_{7K}(a=0,c+1) + \int_{0}^{a} \frac{\left(-14t^{10}+60t^{7}-105t^{4}+140t+81\right)}{162}\,dt \\ =&\gamma_{7K}(a=0,c+1) + \left(-\frac{7a^{11}}{891}+\frac{5a^{8}}{108}-\frac{7a^{5}}{54}+\frac{35a^{2}}{81}+\frac{a}{2}\right)\end{aligned} \tag{B.7.54}$$

Check: When $a = 1$

$$\begin{aligned}\gamma_{8K}(a=1,c) =&\gamma_{7K}(a=0,c+1) + \int_{0}^{1} \frac{\left(-14t^{10}+60t^{7}-105t^{4}+140t+81\right)}{162}\,dt \\ \gamma_{8K}(a=1,c) =&\gamma_{7K}(a=0,c+1) + \frac{37}{44}\end{aligned} \tag{B.7.55}$$

e. Case 2(e) $a \geq 1$:

$$\begin{aligned}\gamma_{9K}(a,c) =&\gamma_{8K}(a=1,c) + \int_{-1}^{a} 1\,dt \\ =&\gamma_{8K}(a=1,c) + (a+1)\end{aligned} \tag{B.7.56}$$

3. **Case 3:** $c \in [-1, 0]$**:** The intersection of the two CDF in this region is $[-1, \infty]$. The cases are below.

a. Case 3(a) $a \in [-1, 0]$:

$$\gamma_{10K}(a, c =) = 0 \tag{B.7.57}$$

b. Case 3(b) $a \in [-1, c]$:

$$\begin{aligned}\gamma_{11K}(a,c) = \int_{c-1}^{a} & \frac{\left(14t^{10}+60t^{7}+105t^{4}+140t+81\right)}{162}\times \\ & \frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^{7}+105\left(t-c\right)^{4}+140\left(t-c\right)+81\right)}{162}\,dt \end{aligned} \tag{B.7.58}$$

$$\begin{aligned} =& -\frac{7a^{21}}{19683}+\frac{49a^{20}c}{13122}-\frac{245a^{19}c^{2}}{13851}+\frac{980a^{18}c^{3}}{19683}-\frac{70a^{17}c\left(49c^{3}-3\right)}{37179}+\frac{7a^{16}c^{2}\left(49c^{3}-20\right)}{2916}+ \\ & \frac{a^{15}\left(-2744c^{6}+4760c^{3}+44\right)}{26244}+\frac{5a^{14}c\left(56c^{6}-350c^{3}-11\right)}{4374}-\frac{35a^{13}c^{2}\left(14c^{6}-308c^{3}-1\right)}{18954}+ \\ & \frac{35a^{12}c^{3}\left(7c^{6}-609c^{3}+201\right)}{39366}-\frac{7a^{11}c\left(14c^{9}-7140c^{6}+13155c^{3}+90\right)}{144342}-\frac{7a^{10}c\left(30c^{7}-234c^{4}-5c-9\right)}{1458}+ \\ & \frac{35a^{9}\left(80c^{9}-2700c^{6}+380c^{3}-486c^{2}+55\right)}{78732}-\frac{5a^{8}\left(14c^{10}-2880c^{7}+3290c^{4}-2268c^{3}+385c-162\right)}{17496}- \\ & \frac{5a^{7}c\left(105c^{7}-574c^{4}+378c^{3}-175c+54\right)}{1458}+\frac{7a^{6}c^{2}\left(175c^{7}-4375c^{4}+3402c^{3}-2975c+1215\right)}{13122}- \\ & \frac{7a^{5}c\left(14c^{9}-2180c^{6}+2268c^{5}-2695c^{3}+1620c^{2}-420\right)}{8748}- \\ & \frac{35a^{4}c\left(70c^{7}-108c^{6}+140c^{4}-135c^{3}+70c-27\right)}{2916}+\frac{35a^{3}\left(c^{3}+1\right)^{2}\left(280c^{3}-729c^{2}+280\right)}{39366}- \\ & \frac{35a^{2}\left(14c^{10}-81c^{9}+60c^{7}-243c^{6}+105c^{4}-243c^{3}+140c-162\right)}{13122}- \\ & \frac{1}{324}a\left(14c^{10}+60c^{7}+105c^{4}+140c-81\right)+ \\ & \frac{\frac{7c^{21}}{138567}+\frac{35c^{18}}{7293}+\frac{38c^{15}}{143}+\frac{1085c^{12}}{33}+\frac{1505c^{9}}{6}+980c^{6}+\frac{9800c^{3}}{3}-6561c}{26244} \end{aligned} \tag{B.7.59}$$

Check: $a = c$

$$\begin{aligned}\gamma_{11K}(a=c,c) = \int_{c-1}^{c} & \frac{\left(14t^{10}+60t^{7}+105t^{4}+140t+81\right)}{162}\times \\ & \frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^{7}+105\left(t-c\right)^{4}+140\left(t-c\right)+81\right)}{162}\,dt \end{aligned} \tag{B.7.60}$$

$$\begin{aligned} =& -\frac{232792560c^{18}-1978736760c^{17}+10553262720c^{16}-38577052800c^{15}}{277134\times 162^{2}}+ \\ & \frac{103825481760c^{14}-209823694080c^{13}+322534091880c^{12}-375785389980c^{11}}{277134\times 162^{2}}+ \\ & \frac{327785092544c^{10}-207120251910c^{9}+89428796820c^{8}-22202968320c^{7}+143555412c^{6}}{277134\times 162^{2}}+ \\ & \frac{1946056266c^{5}-876006495c^{4}+181060880c^{3}-373752225c^{2}-928761002c-365913457}{277134\times 162^{2}} \end{aligned} \tag{B.7.61}$$

c. Case 3(c) $a \in [c, 0]$

$$\begin{aligned}
\gamma_{12K}(a,c) =& \gamma_{11K}(a=c,c) + \int_c^a \frac{\left(14t^{10}+60t^7+105t^4+140t+81\right)}{162} \times \\
& \frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^7+105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\, dt
\end{aligned} \tag{B.7.62}$$

$$\begin{aligned}
=&\gamma_{11K}(a=c,c) + \frac{\left(4938024a^{11}+29099070a^8+81477396a^5+201753552a^2+314269956a\right)c^{10}}{277134\times 162^2} + \\
& \frac{\left(-45265220a^{12}-258658400a^9-678978300a^6-1345023680a^3-1571349780a^2\right)c^9}{277134\times 162^2} + \\
& \frac{\left(188024760a^{13}+1047566520a^{10}+2618916300a^7+4539454920a^4+4714049340a^3\right)c^8}{277134\times 162^2} + \\
& \frac{\left(-465585120a^{14}-2560718160a^{11}-6235515000a^8-10033359336a^5-9428098680a^4-864658080a^2a\right)c^7}{277134\times 162^2} - \\
& \frac{\left(1346871240a\right)c^7}{277134\times 162^2} + \\
& \frac{\left(760455696a^{15}+4209665460a^{12}+10281671400a^9+16159683540a^6+13199338152a^5+4035071040a^3a\right)c^7}{277134\times 162^2} + \\
& \frac{\left(4714049340a^2\right)c^7}{277134\times 162^2} + \\
& \frac{\left(-855512658a^{16}-4888643760a^{13}-12361284936a^{10}-19764088344a^7-13199338152a^6-9078909840a^4\right)c^5}{277134\times 162^2} - \\
& \frac{\left(9428098680a^3\right)c^5}{277134\times 162^2} + \\
& \frac{\left(670990320a^{17}+4073869800a^{14}+10988866980a^{11}+18448810380a^8+9428098680a^7+12716293590a^5\right)c^4}{277134\times 162^2} + \\
& \frac{\left(11785123350a^4+1513151640a^2+2357024670a\right)c^4}{277134\times 162^2} + \\
& \frac{\left(-362121760a^{18}-2405523120a^{15}-7119572460a^{12}-12945852920a^9\right)c^3}{277134\times 162^2} - \\
& \frac{\left(4714049340a^8-12124612500a^6-9428098680a^5-4035071040a^3-4714049340a^2\right)c^3}{277134\times 162^2} + \\
& \frac{\left(128648520a^{19}+960269310a^{16}+3209851260a^{13}+6529831308a^{10}+1571349780a^9+7806864780a^7\right)c^2}{277134\times 162^2} + \\
& \frac{\left(4714049340a^6+4539454920a^4+4714049340a^3\right)c^2}{277134\times 162^2} + \\
& \frac{\left(-27159132a^{20}-232792560a^{17}-906228180a^{14}-2149451304a^{11}-314269956a^{10}-3257017335a^8\right)c^2}{277134\times 162^2} - \\
& \frac{\left(1346871240a^7\right)c^2}{277134\times 162^2} - \\
& \frac{\left(3026303280a^5-2357024670a^4-1498740672a^2-2334576816a\right)c}{277134\times 162^2} + \\
& \frac{2586584a^{21}+25865840a^{18}+120830424a^{15}+358241884a^{12}+57139992a^{11}+723781630a^9+336717810a^8}{277134\times 162^2} + \\
& \frac{1008767760a^6+942809868a^5+999160448a^3+2334576816a^2+1818276174a}{277134\times 162^2} - \\
& \frac{14c^{21}+34884c^{15}+57139992c^{11}-43186715c^9+942809868c^5-499580224c^3+1818276174c}{277134\times 162^2}
\end{aligned} \tag{B.7.63}$$

Check: When $a = 0$

$$\begin{aligned}\gamma_{12K}(a=0,c) =&\gamma_{11K}(a=c,c) + \int_c^0 \frac{\left(14t^{10}+60t^7+105t^4+140t+81\right)}{162}\times \\ &\frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^7+105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt \\ =&\gamma_{11K}(a=c,c) - \frac{14c^{21}+34884c^{15}+57139992c^{11}-43186715c^9+942809868c^5}{277134\times 162^2} + \\ &\frac{499580224c^3+1818276174c}{277134\times 162^2} \end{aligned} \tag{B.7.64}$$

d. Case 3(d) $a \in [0, c+1]$:

$$\begin{aligned}
\gamma_{13K}(a,c) =& \gamma_{12K}(a=0,c) + \int_0^a \frac{\left(-14t^{10}+60t^7-105t^4+140t+81\right)}{162} \times \\
& \frac{\left(-14\left(t-c\right)^{10}+60\left(t-c\right)^7-105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt \\
=& \gamma_{12K}(a=0,c) - \frac{\left(4938024a^{11}+29099070a^8+81477396a^5+201753552a^2+314269956a\right)c^{10}}{277134\times 162^2} + \\
& \frac{\left(-45265220a^{11}-258658400a^8-678978300a^5-1345023680a^2-1571349780a\right)c^9}{277134\times 162^2} + \\
& \frac{\left(188024760a^{12}+1047566520a^9+2618916300a^6+4539454920a^3+4714049340a^2\right)c^8}{277134\times 162^2} + \\
& \frac{\left(-465585120a^{13}-2518392240a^{10}-5986094400a^7-9334981656a^4-9428098680a^3\right)c^7}{277134\times 162^2} + \\
& \frac{\left(864658080a+1346871240\right)c^7}{277134\times 162^2} + \\
& \frac{\left(760455696a^{14}+3938074140a^{11}+8729721000a^8+12085813740a^5+13199338152a^4\right)c^6}{277134\times 162^2} - \\
& \frac{\left(4035071040a^2-4714049340a\right)c^6}{277134\times 162^2} + \\
& \frac{\left(-855512658a^{15}-4136544720a^{12}-8171018856a^9-9288423144a^6-13199338152a^5\right)c^5}{277134\times 162^2} + \\
& \frac{\left(9078909840a^3+9428098680a^2\right)c^5}{277134\times 162^2} + \\
& \frac{\left(670990320a^{16}+2909907000a^{13}+4639978980a^{10}+3171798630a^7+9428098680a^6\right)c^4}{277134\times 162^2} - \\
& \frac{\left(11494132650a^4-11785123350a^3+1513151640a+2357024670\right)c^4}{277134\times 162^2} + \\
& \frac{\left(-362121760a^{17}-1319157840a^{14}-1299758460a^{11}+633713080a^8-4714049340a^7\right)c^3}{277134\times 162^2} + \\
& \frac{\left(8050742700a^5+9428098680a^4-4035071040a^2-4714049340a\right)c^3}{277134\times 162^2} + \\
& \frac{\left(128648520a^{18}+349188840a^{15}-13430340a^{12}-803134332a^9+1571349780a^8-2569032180a^6\right)c^2}{277134\times 162^2} - \\
& \frac{\left(4714049340a^5+4539454920a^3+4714049340a^2\right)c^2}{277134\times 162^2} + \\
& \frac{\left(-27159132a^{19}-41081040a^{16}+91454220a^{13}+146024424a^{10}-314269956a^9+201614985a^7\right)c}{277134\times 162^2} + \\
& \frac{\left(1346871240a^6-1815781968a^4-2357024670a^3+1498740672a+2334576816\right)c}{277134\times 162^2} + \\
& \frac{\left(2586584a^{20}-12193896a^{14}-44803330a^8-336717810a^7-999160448a^2-2334576816a-1818276174\right)}{277134\times 162^2}
\end{aligned} \tag{B.7.65}$$

Check: When $a = c+1$:

$$\begin{aligned}
\gamma_{13K}(a=c+1,c) =& \gamma_{12K}(a=0,c) - \frac{14c^{21}+1330c^{18}+73644c^{15}+6995534c^{12}-458707158c^{10}-2397901935c^9}{277134\times 162^2} - \\
& \frac{-7494129720c^8-17041322376c^7-28280416164c^6-36104783022c^5-37348267320c^4}{277134\times 162^2} + \\
& \frac{-29465995416c^3-17538771888c^2-11646508185c-5543141890}{277134\times 162^2}
\end{aligned} \tag{B.7.66}$$

e. Case 3(e) $a \in [c+1, 1]$:

$$\begin{aligned}\gamma_{14K}(a,c) =&\gamma_{13K}(a=c+1,c) + \int_0^a \frac{\left(-14t^{10}+60t^7-105t^4+140t+81\right)}{162}\,dt\\ =&\gamma_{13K}(a=c+1,c) + \frac{28c^{11}+308c^{10}+1540c^9+4455c^8+7920c^7+8316c^6+4158c^5}{22\times 162}+\\ &\frac{396c^2-2772c-28a^{11}+165a^8-462a^5+1144a^2+1782a-2601}{22\times 162}\end{aligned} \tag{B.7.67}$$

Check: When $a = 1$

$$\gamma_{14K}(a=1,c) =\gamma_{13K}(a=c+1,c) + \frac{\left(28c^{11}+308c^{10}+1540c^9+4455c^8+7920c^7+8316c^6+4158c^5+396c^2-2772c\right)}{22\times 162} \tag{B.7.68}$$

f. Case 3(f) $a \geq 1$:

$$\begin{aligned}\gamma_{15K}(a,c) =&\gamma_{14K}(a=1,c) + \int_1^a 1\,dt\\ =&\gamma_{14K}(a=1,c) + (a-1)\end{aligned} \tag{B.7.69}$$

4. **Case 4:** $c \in [0,1]$**:** The intersection of the two CDF is in the region $[c-1, \infty)$.

a. Case 4(a) $a \leq c-1$:

$$\gamma_{16K}(a,c) = 0 \tag{B.7.70}$$

b. Case 4(b) $a \in [c-1, 0]$:

$$\begin{aligned}\gamma_{17K}(a,c) = &\int_{c-1}^{a} \frac{\left(14t^{10}+60t^{7}+105t^{4}+140t+81\right)}{162} \times \\ &\frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^{7}+105\left(t-c\right)^{4}+140\left(t-c\right)+81\right)}{162}\,dt\end{aligned} \tag{B.7.71}$$

$$\begin{aligned}= &-\frac{14c^{21}-12932920c^{19}+232793225c^{18}-1978736760c^{17}+10486750560c^{16}-38576998536c^{15}}{277134}- \\ &\frac{103825481760c^{14}-209969189430c^{13}+322534532775c^{12}-375728249988c^{11}}{277134}- \\ &\frac{\left(-4938024a^{11}-29099070a^{8}-81477396a^{5}-271591320a^{2}-314269956a+327526434144\right)c^{10}}{277134}- \\ &\frac{\left(45265220a^{12}+258658400a^{9}+678978300a^{6}+1810608800a^{3}+1571349780a^{2}-207316093270\right)c^{9}}{277134}- \\ &\frac{\left(-188024760a^{13}-1047566520a^{10}-2618916300a^{7}-6110804700a^{4}-4714049340a^{3}+89597155725\right)c^{8}}{277134}- \\ &\frac{\left(465585120a^{14}+2539555200a^{11}+6110804700a^{8}+13036383360a^{5}+9428098680a^{4}-22202968320\right)c^{7}}{277134}- \\ &\frac{\left(-760455696a^{15}-4073869800a^{12}-9505696200a^{9}-19011392400a^{6}-13199338152a^{5}+143555412\right)c^{6}}{277134}- \\ &\frac{\left(855512658a^{16}+4512594240a^{13}+10266151896a^{10}+19554575040a^{7}+13199338152a^{6}+2888866134\right)c^{5}}{277134}- \\ &\frac{\left(-670990320a^{17}-3491888400a^{14}-7814422980a^{11}-14476787325a^{8}-9428098680a^{7}-611080470a^{5}\right)c^{4}}{277134}- \\ &\frac{\left(2036934900a^{2}-2357024670a-876006495\right)c^{4}}{277134}- \\ &\frac{\left(362121760a^{18}+1862340480a^{15}+4209665460a^{12}+8018410400a^{9}+4714049340a^{8}+2036934900a^{6}\right)c^{3}}{277134}+ \\ &\frac{\left(5431826400a^{3}+4714049340a^{2}-724243520\right)c^{3}}{277134}- \\ &\frac{\left(-128648520a^{19}-654729075a^{16}-1598210460a^{13}-3491888400a^{10}-1571349780a^{9}-2618916300a^{7}\right)c^{2}}{277134}- \\ &\frac{\left(6110804700a^{4}-4714049340a^{3}-373752225\right)c^{2}}{277134}- \\ &\frac{\left(27159132a^{20}+12932920a^{18}+136936800a^{17}+66512160a^{15}+407386980a^{14}+145495350a^{12}\right)}{277134}+ \\ &\frac{\left(1178071440a^{11}+314269956a^{10}+258658400a^{9}+1987050780a^{8}+4073869800a^{5}+2357024670a^{4}\right)c}{277134}+ \\ &\frac{\left(2715913200a^{2}+3142699560a+889515172\right)c}{277134}+ \\ &\frac{2586584a^{21}-12932920a^{18}-54318264a^{15}-236025790a^{12}-57139992a^{11}-598147550a^{9}-168358905a^{8}}{277134}- \\ &\frac{1357956600a^{6}-942809868a^{5}-1810608800a^{3}-3142699560a^{2}-1818276174a-365913457}{277134}\end{aligned} \tag{B.7.72}$$

Check: When $a = 0$

$$\begin{aligned}\gamma_{17K}(a=0,c) = &-\frac{14c^{21}+34884c^{15}+57139992c^{11}-99994986c^{10}+166511345c^{9}-362619180c^{8}}{277134(162^{2})}- \\ &\frac{298750452c^{6}+471404934c^{5}-340319070c^{4}-452652200c^{3}-283097430c^{2}}{277134(162^{2})}- \\ &\frac{909138087c-364296842}{277134(162^{2})}\end{aligned} \tag{B.7.73}$$

c. Case 4(c) $a \in [0, c]$:

$$\begin{aligned}\gamma_{18K}(a,c) =&\gamma_{17K}(a=0,c)+\\ &\int_0^a \frac{\left(-14t^{10}+60t^7-105t^4+140t+81\right)}{162}\frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^7 105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt\end{aligned} \tag{B.7.74}$$

$$\begin{aligned}=&\gamma_{17K}(a=0,c)+\left(\frac{35a^3\left(c^2\left(280c-729\right)-280\right)\left(c^3-1\right)^2}{39366}+\right.\\ &\frac{a\left(14c^{10}-60c^7+105c^4-140c+81\right)}{324}-\frac{35a^9\left(80c^9+2700c^6+380c^3-486c^2-55\right)}{78732}-\\ &\frac{7a^{11}c\left(14c^9+7140c^6+13155c^3-90\right)}{144342}+\\ &\frac{35a^2\left(c^3\left(14c^7-81c^6-60c^4+243c^3+105c-243\right)-140c+162\right)}{13122}-\\ &\frac{7a^5c\left(c^2\left(14c^7+2180c^4-2268c^3-2695c+1620\right)+420\right)}{8748}+\\ &\frac{5a^8\left(2c^3\left(7c^7+1440c^4+1645c-1134\right)-385c+162\right)}{17496}+\frac{7a^{10}c\left(30c^7+234c^4-5c-9\right)}{1458}-\\ &\frac{a^{15}\left(686c^6+1190c^3-11\right)}{6561}+\frac{5a^{14}c\left(56c^6+350c^3-11\right)}{4374}-\frac{5a^7c\left(7c\left(c^2\left(15c^4+82c-54\right)-25\right)+54\right)}{1458}+\\ &\frac{35a^4\left(c-1\right)c\left(c^2+c+1\right)\left(70c^4-108c^3-70c+27\right)}{2916}+\frac{7a^6c^2\left(7c\left(c^2\left(25c\left(c^3+25\right)-486\right)-425\right)+1215\right)}{13122}+\\ &\frac{35a^{12}c^3\left(7c^3\left(c^3+87\right)+201\right)}{39366}-\frac{35a^{13}c^2\left(14c^3\left(c^3+22\right)-1\right)}{18954}+\frac{7a^{16}c^2\left(49c^3+20\right)}{2916}-\frac{70a^{17}c\left(49c^3+3\right)}{37179}+\\ &\left.\frac{980a^{18}c^3}{19683}-\frac{245a^{19}c^2}{13851}+\frac{49a^{20}c}{13122}-\frac{7a^{21}}{19683}\right)\end{aligned} \tag{B.7.75}$$

Check:When $a = c$

$$\begin{aligned}\gamma_{18K}(a=c,c) =&\gamma_{17K}(a=0,c)+\int_0^c \frac{\left(-14t^{10}+60t^7-105t^4+140t+81\right)}{162}\times\\ &\frac{\left(14\left(t-c\right)^{10}+60\left(t-c\right)^7 105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt\\ =&\gamma_{K3}(a=0,c)-\frac{14c^{21}-12932920c^{19}-665c^{18}+66512160c^{16}+54264c^{15}-145495350c^{13}-8670935c^{12}}{277134\times 162^2}+\\ &\frac{258658400c^{10}+205541050c^9-168358905c^8-271591320c^6+905304400c^3-1818276174c}{277134\times 162^2}\end{aligned} \tag{B.7.76}$$

d. Case 4(d) $a \in [c, 1]$:

$$\begin{aligned}\gamma_{19K}(a,c) =&\gamma_{18K}(a=c,c)+\\ &\int_c^a \frac{\left(-14t^{10}+60t^7-105t^4+140t+81\right)}{162}\frac{\left(-14\left(t-c\right)^{10}+60\left(t-c\right)^7-105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt\end{aligned} \tag{B.7.77}$$

$$\begin{aligned}=&\frac{7a^{21}}{19683}-\frac{49a^{20}c}{13122}+\frac{245a^{19}c^2}{13851}-\frac{70a^{18}\left(14c^3+1\right)}{19683}+\\ &\frac{70a^{17}c\left(49c^3+17\right)}{37179}-\frac{7a^{16}c^2\left(49c^3+55\right)}{2916}+\frac{a^{15}\left(686c^6+2170c^3+109\right)}{6561}-\frac{5a^{14}c\left(56c^6+490c^3+109\right)}{4374}+\\ &\frac{35a^{13}c^2\left(14c^6+364c^3+239\right)}{18954}-\frac{35a^{12}\left(7c^9+651c^6+1101c^3+59\right)}{39366}+\\ &\frac{7a^{11}\left(14c^{10}+7260c^7+31155c^4+6490c-162\right)}{144342}-\frac{7a^{10}c\left(30c^7+354c^4+205c-9\right)}{1458}+\\ &\frac{35a^9\left(80c^9+3180c^6+4580c^3-486c^2+265\right)}{78732}-\frac{5a^8\left(14c^{10}+3000c^7+10640c^4-2268c^3+1855c-162\right)}{17496}+\\ &\frac{5a^7c\left(105c^7+994c^4-378c^3+385c-54\right)}{1458}-\frac{7a^6\left(175c^9+5425c^6-3402c^5+4025c^3-1215c^2+350\right)}{13122}+\\ &\frac{7a^5\left(14c^{10}+2300c^7-2268c^6+2905c^4-1620c^3+700c-162\right)}{8748}-\\ &\frac{35a^4c\left(70c^7-108c^6+140c^4-135c^3+70c-27\right)}{2916}+\frac{35a^3\left(c^3+1\right)^2\left(280c^3-729c^2+280\right)}{39366}-\\ &\frac{35a^2\left(14c^{10}-81c^9+60c^7-243c^6+105c^4-243c^3+140c-162\right)}{13122}-\\ &\frac{1}{324}a\left(14c^{10}+60c^7+105c^4+140c-81\right)+\\ &\frac{-\frac{7c^{21}}{138567}-\frac{18c^{15}}{143}+\frac{2268c^{11}}{11}+\frac{1295c^9}{6}+3402c^5+\frac{9800c^3}{3}-6561c}{26244}\end{aligned} \tag{B.7.78}$$

Check: When $a = 1$:

$$\begin{aligned}\gamma_{19K}(a=1,c) =&\gamma_{18K}(a=c,c)-\frac{14c^{21}+34884c^{15}-57139992c^{11}+528544926c^{10}-2976188215c^9}{277134\times 162^2}-\\ &\frac{9065479500c^8-16162454880c^7+17269328076c^6-8956693746c^5-340319070c^4}{277134\times 162^2}-\\ &\frac{-452652200c^3+2859602130c^2+4545690435c-5323231862}{277134\times 162^2}\end{aligned} \tag{B.7.79}$$

e. Case 4(e) $a \in [1, c+1]$:

$$\gamma_{20K}(a,c) =\gamma_{19K}(a=1,c)+\int_1^a \frac{\left(-14\left(t-c\right)^{10}+60\left(t-c\right)^7-105\left(t-c\right)^4+140\left(t-c\right)+81\right)}{162}\,dt \tag{B.7.80}$$

$$\begin{aligned}=&\gamma_{19K}(a=1,c)-\frac{\left(308a-308\right)c^{10}+\left(1540-1540a^2\right)c^9+\left(4620a^3-4620\right)c^8+\left(-9240a^4+1320a+7920\right)c^7}{3564}-\\ &\frac{\left(12936a^5-4620a^2-8316\right)c^6+\left(-12936a^6+9240a^3+3696\right)c^5+\left(9240a^7-11550a^4+220a+2090\right)c^4}{3564}-\\ &\frac{\left(-4620a^8+9240a^5-440a^2-4180\right)c^3+\left(1540a^9-4620a^6+440a^3+2640\right)c^2}{3564}-\\ &\frac{\left(-308a^{10}+1320a^7-220a^4+3080a-3872\right)c+28a^{11}-165a^8+44a^5-1540a^2-1782a+3415}{3564}\end{aligned} \tag{B.7.81}$$

Check: When $a = c + 1$

$$\gamma_{20K}(a = c+1, c) = \gamma_{19K}(a = 1, c) - \frac{7c^{11}}{891} + \frac{7c^{10}}{81} - \frac{35c^9}{81} + \frac{5c^8}{4} - \frac{20c^7}{9} + \frac{7c^6}{3} - \frac{7c^5}{6} + c \quad \text{(B.7.82)}$$

f. Case 4(f) $a \geq c + 1$:

$$\begin{aligned}\gamma_{21K}(a, c) =& \gamma_{20K}(a = c+1, c) + \int_{c+1}^{a} \\ =& \gamma_{20K}(a = c+1, c) + a - c - 1\end{aligned} \quad \text{(B.7.83)}$$

5. **Case 5:** $c \in [1, 2]]$**:** The intersection of the two CDF is the region $[0, \infty]$. The cases are below.

a. Case 5(a) $a \leq c - 1$:

$$\gamma_{22K}(a, c) = 0 \quad \text{(B.7.84)}$$

b. Case 5(b) $a \in [c-1, 0]$:

$$
\begin{aligned}
\gamma_{23K}(a,c) = & \int_{-1}^{a} \frac{81 + 140t - 105t^4 + 60t^7 - 14t^10}{162} \frac{81 + 140(t-c) + 105(t-c)^4 + 60(t-c)^7 + 14(t-c)^10}{162} \, dt \\
= & \frac{\left(4938024a^{11} + 29099070a^8 + 81477396a^5 + 271591320a^2 + 314269956a\right) c^{10}}{277134 \times 162^2} + \\
& \frac{\left(-45265220a^{12} - 258658400a^9 - 678978300a^6 - 1810608800a^3 - 1571349780a^2\right) c^9}{277134 \times 162^2} + \\
& \frac{\left(188024760a^{13} + 1047566520a^{10} + 2618916300a^7 + 6110804700a^4 + 4714049340a^3\right) c^8}{277134 \times 162^2} + \\
& \frac{\left(-465585120a^{14} - 2560718160a^{11} - 6235515000a^8 - 13385572200a^5 - 9428098680a^4\right) c^7}{277134 \times 162^2} - \\
& \frac{\left(1163962800a^2 - 1346871240a\right) c^7}{277134 \times 162^2} + \\
& \frac{\left(760455696a^{15} + 4209665460a^{12} + 10281671400a^9 + 21048327300a^6 + 13199338152a^5 + 5431826400a^3\right) c^6}{277134 \times 162^2} + \\
& \frac{\left(4714049340a^2\right) c^6}{277134 \times 162^2} + \\
& \frac{\left(-855512658a^{16} - 4888643760a^{13} - 12361284936a^{10} - 24792407640a^7 - 13199338152a^6\right) c^5}{277134 \times 162^2} - \\
& \frac{\left(12221609400a^4 - 9428098680a^3\right) c^5}{277134 \times 162^2} + \\
& \frac{\left(670990320a^{17} + 4073869800a^{14} + 10988866980a^{11} + 22115293200a^8 + 9428098680a^7\right) c^4}{277134 \times 162^2} + \\
& \frac{\left(16906559670a^5 + 11785123350a^4 + 2036934900a^2 + 2357024670a\right) c^4}{277134 \times 162^2} + \\
& \frac{\left(-362121760a^{18} - 2405523120a^{15} - 7119572460a^{12} - 14808193400a^9 - 4714049340a^8\right) c^3}{277134 \times 162^2} - \\
& \frac{\left(15616500900a^6 - 9428098680a^5 - 5431826400a^3 - 4714049340a^2\right) c^3}{277134 \times 162^2} + \\
& \frac{\left(128648520a^{19} + 960269310a^{16} + 3209851260a^{13} + 7158371220a^{10} + 1571349780a^9\right) c^2}{277134 \times 162^2} + \\
& \frac{\left(9602693100a^7 + 4714049340a^6 + 6110804700a^4 + 4714049340a^3\right) c^2}{277134 \times 162^2} + \\
& \frac{\left(-27159132a^{20} - 232792560a^{17} - 906228180a^{14} - 2289126840a^{11} - 314269956a^{10} - 3855626775a^8\right) c}{277134 \times 162^2} - \\
& \frac{\left(1346871240a^7 - 4073869800a^5 - 2357024670a^4 - 2715913200a^2 - 3142699560a\right) c}{277134 \times 162^2} + \\
& \frac{2586584a^{21} + 25865840a^{18} + 120830424a^{15} + 381521140a^{12} + 57139992a^{11} + 856805950a^9}{277134 \times 162^2} + \\
& \frac{336717810a^8 + 1357956600a^6 + 942809868a^5 + 1810608800a^3 + 3142699560a^2 + 1818276174a}{277134 \times 162^2} - \\
& \frac{14c^{21} + 34884c^{15} + 57139992c^{11} - 99994986c^{10} + 166511345c^9 - 362619180c^8 + 298750452c^6}{277134 \times 162^2} + \\
& \frac{471404934c^5 - 340319070c^4 - 452652200c^3 - 283097430c^2 + 909138087c - 364296842}{277134 \times 162^2}
\end{aligned}
\tag{B.7.85}
$$

Check: When $a = 0$

$$\begin{aligned}\gamma_{23K}(a=0,c) =& -\frac{14c^{21}+27166884c^{15}-194434695c^{12}+57139992c^{11}-99994986c^{10}+792141350c^{9}}{277134\times 162^2}- \\ & \frac{-362619180c^{8}-1059206148c^{6}+471404934c^{5}+409643325c^{4}+1584282700c^{3}}{277134\times 162^2}- \\ & \frac{-3063177810c^{2}+1801950462c-364296842}{277134\times 162^2}\end{aligned} \tag{B.7.86}$$

c. Case 5(c) $a \in [0,1]$:

$$\begin{aligned}\gamma_{24K}(a,c) =& \gamma_{23K}(a=0,c) + \int_1^a \frac{81+140(t)+105(t)^4+60(t)^7+14(t)^{10}}{162}\times \\ & \frac{81+140(t-c)-105(t-c)^4+60(t-c)^7-14(t-c)^{10}}{162}\,dt \\ =& \gamma_{23K}(a=1,c) - -\frac{\left(4938024a^{11}+29099070a^{8}+81477396a^{5}+271591320a^{2}+314269956a\right)c^{10}}{277134\times 162^2}- \\ & \frac{\left(-45265220a^{12}-258658400a^{9}-678978300a^{6}-1810608800a^{3}-1571349780a^{2}\right)c^{9}}{277134\times 162^2}- \\ & \frac{\left(188024760a^{13}+1047566520a^{10}+2618916300a^{7}+6110804700a^{4}+4714049340a^{3}\right)c^{8}}{277134\times 162^2}- \\ & \frac{\left(-465585120a^{14}-2518392240a^{11}-5986094400a^{8}-12687194520a^{5}-9428098680a^{4}+1163962800a^{2}\right)c^{7}}{277134\times 162^2}+ \\ & \frac{(1346871240a)\,c^{7}}{277134\times 162^2}- \\ & \frac{\left(760455696a^{15}+3938074140a^{12}+8729721000a^{9}+16974457500a^{6}+13199338152a^{5}\right)c^{6}}{277134\times 162^2}- \\ & \frac{\left(5431826400a^{3}-4714049340a^{2}\right)c^{6}}{277134\times 162^2}- \\ & \frac{\left(-855512658a^{16}-4136544720a^{13}-8171018856a^{10}-14316742440a^{7}-13199338152a^{6}\right)c^{5}}{277134\times 162^2}+ \\ & \frac{\left(12221609400a^{4}+9428098680a^{3}\right)c^{5}}{277134\times 162^2}- \\ & \frac{\left(670990320a^{17}+2909907000a^{14}+4639978980a^{11}+6838281450a^{8}+9428098680a^{7}\right)c^{4}}{277134\times 162^2}- \\ & \frac{\left(15684398730a^{5}-11785123350a^{4}+2036934900a^{2}+2357024670a\right)c^{4}}{277134\times 162^2}- \\ & \frac{\left(-362121760a^{18}-1319157840a^{15}-1299758460a^{12}-1228627400a^{9}-4714049340a^{8}\right)c^{3}}{277134\times 162^2}+ \\ & \frac{\left(11542631100a^{6}+9428098680a^{5}-5431826400a^{3}-4714049340a^{2}\right)c^{3}}{277134\times 162^2}- \\ & \frac{\left(128648520a^{19}+349188840a^{16}-13430340a^{13}-174594420a^{10}+1571349780a^{9}\right)c^{2}}{277134\times 162^2}- \\ & \frac{\left(4364860500a^{7}-4714049340a^{6}+6110804700a^{4}+4714049340a^{3}\right)c^{2}}{277134\times 162^2}- \\ & \frac{\left(-27159132a^{20}-41081040a^{17}+91454220a^{14}+31744440a^{11}-314269956a^{10}+800224425a^{8}\right)c}{277134\times 162^2}+ \\ & \frac{\left(1346871240a^{7}-2444321880a^{5}-2357024670a^{4}+2715913200a^{2}+3142699560a\right)c}{277134\times 162^2}- \\ & \frac{2586584a^{21}-12193896a^{15}-177827650a^{9}-336717810a^{8}-1810608800a^{3}-3142699560a^{2}-1818276174a}{277134\times 162^2}\end{aligned} \tag{B.7.87}$$

Check: When $a = 1$

$$\begin{aligned}\gamma_{25K}(a=1,c) =&\gamma_{23K}(a=1,c) - \frac{701375766c^{10} - 4364860500c^{9} + 14679361620c^{8} - 28574530920c^{7} + 33456170748c^{6}}{277134 \times 162^{2}} - \\ &\frac{-19029448746c^{5} + 1411693920c^{4} + 1901139240c^{3} + 3607106580c^{2} + 2945050407c - 7295737306}{277134 \times 162^{2}}\end{aligned} \tag{B.7.88}$$

d. Case 5(d) $a \in [1, c]$:

$$\begin{aligned}\gamma_{25K}(a,c) =&\gamma_{25K}(a=1,c) + \int_{1}^{a} \frac{81 + 140(t-c) - 105(t-c)^{4} + 60(t-c)^{7} - 14(t-c)^{10}}{162} \\ =&\gamma_{25K}(a=c,c) + \frac{28c^{11} - 308ac^{10} + 1540a^{2}c^{9} + \left(165 - 4620a^{3}\right)c^{8} + \left(9240a^{4} - 1320a\right)c^{7}}{22 \times 162} + \\ &\frac{\left(4620a^{2} - 12936a^{5}\right)c^{6} + \left(12936a^{6} - 9240a^{3}\right)c^{5} + \left(-9240a^{7} + 11550a^{4} - 2310a\right)c^{4}}{22 \times 162} + \\ &\frac{\left(4620a^{8} - 9240a^{5} + 4620a^{2}\right)c^{3} + \left(-1540a^{9} + 4620a^{6} - 4620a^{3}\right)c^{2}}{22 \times 162} + \\ &\frac{\left(308a^{10} - 1320a^{7} + 2310a^{4} - 3080a\right)c - 28a^{11} + 165a^{8} - 462a^{5} + 1540a^{2} + 1782a}{22} - \frac{28c^{11} + 165c^{8}}{22 \times 162}\end{aligned} \tag{B.7.89}$$

Check: when $a = c$

$$\gamma_{25K}(a=c,c) =\gamma_{25K}(a=1,c) - \frac{c\left(28c^{10} + 165c^{7} + 462c^{4} + 1540c - 1782\right)}{22} \tag{B.7.90}$$

e. Case 5(e) $a \in [c, c+1]$

$$\begin{aligned}\gamma_{26K}(a,c) =&\gamma_{25K}(a=c,c) + \int_{c}^{a} \frac{81 + 140(t-c) + 105(t-c)^{4} + 60(t-c)^{7} + 14(t-c)^{10}}{162} \\ =&\gamma_{25K}(a=c,c) - \frac{28c^{11} - 308ac^{10} + 1540a^{2}c^{9} + \left(-4620a^{3} - 165\right)c^{8} + \left(9240a^{4} + 1320a\right)c^{7}}{22 \times 162} - \\ &\frac{\left(-12936a^{5} - 4620a^{2}\right)c^{6} + \left(12936a^{6} + 9240a^{3} + 462\right)c^{5}}{22 \times 162} - \\ &\frac{\left(-9240a^{7} - 11550a^{4} - 2310a\right)c^{4} + \left(4620a^{8} + 9240a^{5} + 4620a^{2}\right)c^{3}}{22 \times 162} - \\ &\frac{\left(-1540a^{9} - 4620a^{6} - 4620a^{3} - 1540\right)c^{2} + \left(308a^{10} + 1320a^{7} + 2310a^{4} + 3080a + 1782\right)c}{22 \times 162} - \\ &\frac{-28a^{11} - 165a^{8} - 462a^{5} - 1540a^{2} - 1782a}{22 \times 162}\end{aligned} \tag{B.7.91}$$

Check: When $a = c + 1$

$$\gamma_{26K}(a=c+1,c) =\gamma_{25K}(a=c,c) + \frac{3977}{22 \times 162} \tag{B.7.92}$$

f. Case 5(f) $a \geq c+1$

$$\begin{aligned}\gamma_{27K}(a,c) =&\gamma_{26K}(a=c+1,c)+\int_{c+1}^{a} 1\ dt\\ =&\gamma_{26K}(a=c+1,c)+(a-c-1)\end{aligned} \tag{B.7.93}$$

6. **Case 6:** $c \geq 2$**:**
a. Case 6(a) $a \leq c-1$:

$$\gamma_{28K}(a,c)=0 \tag{B.7.94}$$

b. Case 6(b) $a \in [c-1,1]$:

$$\begin{aligned}\gamma_{29K}(a,c)=&\int_{c-1}^{a} \frac{81+140(t-c)+105(t-c)^4+60(t-c)^7+14(t-c)^{10}}{162}\\ =&\frac{1}{162}\left(70a^2+\frac{14}{11}(a-c)^{11}+\frac{15}{2}(a-c)^8+21(a-c)^5-140ac+81a+70c^2-81c+\frac{567}{22}\right)\end{aligned} \tag{B.7.95}$$

c. Case 6(c) $a \in [c,c+1]$:

$$\begin{aligned}\gamma_{30K}(a,c)=&\int_{c-1}^{c} \frac{81+140(t-c)+105(t-c)^4+60(t-c)^7+14(t-c)^{10}}{162}+\\ &\int_{c}^{a} \frac{81+140(t-c)-105(t-c)^4+60(t-c)^7-14(t-c)^{10}}{162}\\ =&\frac{7}{44}+\frac{1}{162}\left(70a^2-\frac{1}{11}14(a-c)^{11}+\frac{15}{2}(a-c)^8-21(a-c)^5-140ac+81a+70c^2-81c\right)\end{aligned} \tag{B.7.96}$$

d. Case 6(d) $a \geq c+1$:

$$\begin{aligned}\gamma_{31K}(a,c)=&\int_{c-1}^{c} \frac{81+140(t-c)+105(t-c)^4+60(t-c)^7+14(t-c)^{10}}{162}\\ &\int_{c}^{c+1} \frac{81+140(t-c)-105(t-c)^4+60(t-c)^7-14(t-c)^{10}}{162}+a-c-1\\ =&a-c\end{aligned} \tag{B.7.97}$$

□

Derivation B.7.5. *Let K be a Tricube kernel as in Def B.7.1 with the CDF ν_K/ The L2-nor of the CCDF, $(1-\nu_K)$ at two central points 0 and $c \in \mathbb{R}$ as in Eqn (5.4.25) is taking the negatives of a and c in Deriv B.7.4.*

B.8 Logistic kernel

Definition B.8.1. *Let the Logistic kernel be defined as*

$$K(u) = \frac{1}{2 + e^u + e^{-u}} \tag{B.8.1}$$

Derivation B.8.1. *Let K be a Logistic kernel as in Definition B.8.1. The cdf for the Logistic kernel is*

$$\nu_K(t) = \frac{\mathrm{e}^t}{1 + \mathrm{e}^t} \tag{B.8.2}$$

Proof. Let K be a Logistic kernel as in Definition B.8.1. The cdf is the integration of the kernel function,

$$\begin{aligned} \nu_K(t) &= \int_{-\infty}^{t} \frac{1}{2 + e^u + \mathrm{e}^{-u}}\, du \\ &= \frac{1}{2} \int_{-\infty}^{t} \frac{\operatorname{sech}^2(\frac{u}{2})}{2}\, dt \end{aligned} \tag{B.8.3}$$

Applying substitution by parts,

$$v = \frac{u}{2} \rightarrow \frac{dv}{du} = \frac{1}{2} \rightarrow du = 2v \tag{B.8.4}$$

Then,

$$\nu_K(t) = \frac{1}{2} \int_{-\infty}^{t/2} \operatorname{sech}^2(v) dv = \frac{1}{2} \left[\tanh(v)\right]_{-\infty}^{t/2} = \frac{e^t}{1 + e^t} \tag{B.8.5}$$

□

Derivation B.8.2. *Let K be a Logistic kernel as in Definition B.8.1. The partial L2-product of Logistic kernels at two different central points 0 and $c \in \mathbb{R}$ when the integration boundary is $(-\infty, \infty)$ is*

$$\lambda_K(c) = \frac{(c-2)\,\mathrm{e}^{2c} + (c+2)\,\mathrm{e}^{c}}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} \tag{B.8.6}$$

Proof. Suppose we have a Logistic kernel K as in Definition B.8.1. The partial

L2-product of the Logistic kernel as in Eqn (B.3.3) is

$$\begin{aligned}\lambda_K(a,c) =& \int_{-\infty}^{a} \frac{1}{2+(\mathrm{e}^{u}+\mathrm{e}^{-u})} \frac{1}{2+(\mathrm{e}^{u-c}+\mathrm{e}^{-(u-c)})}\,du \\ =& \int \frac{1}{4+2\mathrm{e}^{u-c}+2\mathrm{e}^{-u+c}+2\mathrm{e}^{e}+\mathrm{e}^{2u-c}+\mathrm{e}^{c}+2\mathrm{e}^{-u}+\mathrm{e}^{-c}+\mathrm{e}^{-2u+c}}\,du\end{aligned} \tag{B.8.7}$$

Multiply by $\frac{\mathrm{e}^c}{\mathrm{e}^c}$

$$= \int \frac{\mathrm{e}^{c}}{(\mathrm{e}^{u}+\mathrm{e}^{2c-u}+2\mathrm{e}^{c})(\mathrm{e}^{u}+\mathrm{e}^{-u}+2)}\,du \tag{B.8.8}$$

Since e^c is a constant, we take it out simplifying the integration to

$$=e^{c}\int \frac{1}{(\mathrm{e}^{u}+e^{2c-u}+2\mathrm{e}^{c})(\mathrm{e}^{u}+\mathrm{e}^{-u}+2)}\,du$$

and substitute the following, $w = \mathrm{e}^u$, and $\frac{dw}{du} = \mathrm{e}^u = w$ and $du = \frac{1}{w}\,dw$ and we have $e^{2u} = \mathrm{e}^u.\mathrm{e}^u = w^2$. The inside of the integral is

$$= \int \frac{1}{(w+\frac{\mathrm{e}^{2c}}{w}+2\mathrm{e}^{c})(w+\frac{1}{w}+2)}\frac{1}{w}\,dw \tag{B.8.9}$$

$$= \int \frac{w}{(w+\mathrm{e}^{c})^{2})w+1)^{2}}\,dw. \tag{B.8.10}$$

Applying partial fraction to above, and letting $\alpha = e^c$

$$\begin{aligned}\frac{w}{(w+\mathrm{e}^{c})^{2})w+1)^{2}} =& \frac{A}{w+\alpha}+\frac{B}{(w+\alpha)^{2}}+\frac{D}{1+w}+\frac{E}{(1+w)^{2}} \\ w =& A(w+\alpha)(1+w)^{2}+B(1+w)^{2}+D(w+\alpha)^{2}(1+w)+ \\ & E(w+\alpha)^{2} \\ A =& \frac{-(\alpha+1)}{(1-\alpha)^{2}(\alpha-1)} \\ B =& \frac{-\alpha}{(1-\alpha)^{2}} \\ D =& \frac{(1+\alpha)}{(1-\alpha)^{2}(\alpha-1)} \\ E =& \frac{-1}{(\alpha-1)^{2}}\end{aligned}$$

Then, we can obtain

$$
\begin{aligned}
&\int \frac{w}{(w+\mathrm{e}^c)^2)w+1)^2}\,dw \\
=&\int \frac{-(\mathrm{e}^c+1)}{(1-\mathrm{e}^c)^2(\mathrm{e}^c-1)(w+\mathrm{e}^c)} - \int \frac{\mathrm{e}^c}{(1-\mathrm{e}^c)^2(w+\mathrm{e}^c)^2} + \int \frac{1+\mathrm{e}^c}{(1-\mathrm{e}^c)^2(\mathrm{e}^c-1)(w+1)} - \\
&\int \frac{1}{(\mathrm{e}^c-1)^2(1+w)^2}\,dw \\
=&\frac{-(\mathrm{e}^c+1)}{(1-\mathrm{e}^c)^2(\mathrm{e}^c-1)}\int \frac{1}{(w+\mathrm{e}^c)}\,dw - \frac{\mathrm{e}^c}{(1-\mathrm{e}^c)^2}\int \frac{1}{(w+\mathrm{e}^c)^2}\,dw + \\
&\frac{1+\mathrm{e}^c}{(1-\mathrm{e}^c)^2(\mathrm{e}^c-1)}\int \frac{1}{(w+1)}\,du - \\
&\frac{1}{(\mathrm{e}^c-1)^2}\int \frac{1}{(1+w)^2}\,dw.
\end{aligned}
$$

By integrating each terms above w.r.t. w and plug-in back $w = \mathrm{e}^u$ we obtain the result of integration is

$$
\frac{(-\mathrm{e}^c-1)\,\mathrm{e}^c\ln(\mathrm{e}^u+\mathrm{e}^c)+\mathrm{e}^c(\mathrm{e}^c+1)\ln(\mathrm{e}^u+1)}{\mathrm{e}^{3c}-3\mathrm{e}^{2c}+3\mathrm{e}^c-1} + \frac{\mathrm{e}^{2c}}{(\mathrm{e}^{2c}-2\mathrm{e}^c+1)(\mathrm{e}^u+\mathrm{e}^c)} + \frac{\mathrm{e}^c}{(\mathrm{e}^{2c}-2\mathrm{e}^c+1)(\mathrm{e}^u+1)} \tag{B.8.11}
$$

Now, we can find the limit of the integration from $-\infty$ to ∞

$$
\left[\frac{(\mathrm{e}^c+1)\mathrm{e}^c}{\mathrm{e}^{3c}-3\mathrm{e}^{2c}+3\mathrm{e}^c-1}\ln\left(\frac{\mathrm{e}^u+1}{\mathrm{e}^u+\mathrm{e}^c}\right)\right]_{-\infty}^{\infty} + \left[\frac{\mathrm{e}^{2c}}{(\mathrm{e}^{2c}-2\mathrm{e}^c+1)(\mathrm{e}^u+\mathrm{e}^c)}\right]_{-\infty}^{\infty} + \left[\frac{\mathrm{e}^c}{(\mathrm{e}^{2c}-2\mathrm{e}^c+1)(\mathrm{e}^u+1)}\right]_{\infty}^{\infty} \tag{B.8.12}
$$

For the first in the bracket, the limit of $u \to \infty$ for $\ln \frac{\mathrm{e}^u+1}{\mathrm{e}^u+\mathrm{e}^c} \to 0$. For the second term, as $u \to \infty$, $\frac{1}{\mathrm{e}^c+\mathrm{e}^c} \to 0$ and the third term will also go to 0 because as $u \to \infty$ the term $\frac{1}{\mathrm{e}^u+1} \to 0$. Therefore, the integration

$$
\int_{\infty}^{\infty} \mathrm{e}^c \frac{1}{(e^u+e^{2c-u}+2e^c)(e^u+e^{-u}+2)}\,du = \frac{c\mathrm{e}^c(\mathrm{e}^c+1)}{\mathrm{e}^{3c}-3\mathrm{e}^{2c}+3\mathrm{e}^c-1} - \frac{2\mathrm{e}^c}{\mathrm{e}^{2c}-2\mathrm{e}^c+1} \tag{B.8.13}
$$

since $\ln \mathrm{e}^{-c} = -c$

$$\lambda_K(c) = \frac{(c-2)\,\mathrm{e}^{2c} + (c+2)\,\mathrm{e}^{c}}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} \tag{B.8.14}$$

□

Derivation B.8.3. *Let K be a Logistic kernel as in Definition B.8.1. The partial L2-product of Logistic kernels at two different central points 0 and $c \in \mathbb{R}$ when the integration boundary is $(-\infty, a]$ where $a \in \mathbb{R}$ is*

$$\begin{aligned}\lambda_K(a,c) =& \frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\left(\ln(\mathrm{e}^{-a}+1) - \ln(\mathrm{e}^{c-a}+1) + c\right)}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} - \\ & \left(\frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-a} + 2\mathrm{e}^{c}}{(\mathrm{e}^{3c} - \mathrm{e}^{2c} - \mathrm{e}^{c} + 1)\,\mathrm{e}^{-a} + (\mathrm{e}^{3c} - 2\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-2a} + \mathrm{e}^{2c} - 2\mathrm{e}^{c} + 1}\right)\end{aligned} \tag{B.8.15}$$

Proof. Suppose we have a Logistic kernel K as in Definition B.8.1. The partial L2-product of the Logistic kernel as in Eqn (B.3.3) is

$$\lambda_K(a,c) = \int_{-\infty}^{a} \frac{1}{2 + (\mathrm{e}^{u} + \mathrm{e}^{-u})} \frac{1}{2 + (\mathrm{e}^{u-c} + \mathrm{e}^{-(u-c)})}\, du \tag{B.8.16}$$

$$= \int \frac{1}{4 + 2\mathrm{e}^{u-c} + 2\mathrm{e}^{-u+c} + 2\mathrm{e}^{e} + \mathrm{e}^{2u-c} + \mathrm{e}^{c} + 2\mathrm{e}^{-u} + \mathrm{e}^{-c} + \mathrm{e}^{-2u+c}}\, du \tag{B.8.17}$$

Using Eqn B.8.11, and add the boundary from $-\infty$ to $a \in \mathbb{R}$

$$\begin{aligned}\lambda_K(a,c) = \Bigg[& \frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\ln(\mathrm{e}^{-u}+1)}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} - \frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\ln(\mathrm{e}^{c-u}+1)}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} - \\ & \frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-u} + 2\mathrm{e}^{c}}{(\mathrm{e}^{3c} - \mathrm{e}^{2c} - \mathrm{e}^{c} + 1)\,\mathrm{e}^{-u} + (\mathrm{e}^{3c} - 2\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-2u} + \mathrm{e}^{2c} - 2\mathrm{e}^{c} + 1}\Bigg]_{-\infty}^{a}\end{aligned} \tag{B.8.18}$$

$$\begin{aligned}=& \frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\left(\ln(\mathrm{e}^{-a}+1) - \ln(\mathrm{e}^{c-a}+1) + c\right)}{\mathrm{e}^{3c} - 3\mathrm{e}^{2c} + 3\mathrm{e}^{c} - 1} - \\ & \left(\frac{(\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-a} + 2\mathrm{e}^{c}}{(\mathrm{e}^{3c} - \mathrm{e}^{2c} - \mathrm{e}^{c} + 1)\,\mathrm{e}^{-a} + (\mathrm{e}^{3c} - 2\mathrm{e}^{2c} + \mathrm{e}^{c})\,\mathrm{e}^{-2a} + \mathrm{e}^{2c} - 2\mathrm{e}^{c} + 1}\right).\end{aligned} \tag{B.8.19}$$

Note that:

$$\lim_{u\to-\infty} \frac{\left(\mathrm{e}^{2c}+\mathrm{e}^{c}\right)\mathrm{e}^{-u}+2\mathrm{e}^{c}}{\left(\mathrm{e}^{3c}-\mathrm{e}^{2c}-\mathrm{e}^{c}+1\right)\mathrm{e}^{-u}+\left(\mathrm{e}^{3c}-2\mathrm{e}^{2c}+\mathrm{e}^{c}\right)\mathrm{e}^{-2u}+\mathrm{e}^{2c}-2\mathrm{e}^{c}+1} = 0 \tag{B.8.20}$$

□

Derivation B.8.4. *Let K be a Logistic kernel as in Definition B.8.1. The partial L2-product of the Logistic cdfs at different central points 0 and $c \in \mathbb{R}$ from $(-\infty, a]$ where $a \in \mathbb{R}$ is*

$$\gamma_K(a,c) = \begin{cases} \ln(\mathrm{e}^{a}+1) - a - \frac{1}{\mathrm{e}^{a}+1} & \text{if} \quad c=0 \\ \frac{\mathrm{e}^{c}\left(\ln\left(\frac{\mathrm{e}^{a}+\mathrm{e}^{c}}{\mathrm{e}^{c}}\right)\right)-\ln(\mathrm{e}^{a}+1)}{\mathrm{e}^{c}-1} & \text{otherwise} \end{cases}$$

Proof. Let K be a Logistic kernel as in Def B.8.1 with CDF as in B.8.1. The partial L2-product of the CDF as in Eqn (5.4.3) is

$$\begin{aligned} \gamma_K(a,c) &= \int_{-\infty}^{a} \frac{\mathrm{e}^{t}}{1+\mathrm{e}^{t}} \frac{\mathrm{e}^{t-c}}{1+\mathrm{e}^{t-c}}\, dt \\ &= \int_{-\infty}^{a} \frac{\mathrm{e}^{2t}}{(\mathrm{e}^{t}+1)(\mathrm{e}^{t}+\mathrm{e}^{c})}\, dt \end{aligned} \tag{B.8.21}$$

substitute $w = \mathrm{e}^{t}+1$ and $\frac{dw}{dt} = \mathrm{e}^{t} = w$ and $dt = \frac{1}{w-1}\, dw$ with $\mathrm{e}^{2t} = (w-1)^2$ to the above to get

$$\begin{aligned} &= \int \frac{(w-1)^2}{w(w-1+\mathrm{e}^{c})} \frac{1}{w-1}\, dw \\ &= \int \frac{(w-1)}{w(w-1+\mathrm{e}^{c})}\, dw \end{aligned}$$

Apply partial fraction,

$$\frac{(w-1)}{w(w-1+\mathrm{e}^{c})} = \frac{A}{w} + \frac{B}{(w-1+\mathrm{e}^{c}}$$

where

$$\begin{aligned} A =& \frac{-1}{\mathrm{e}^{c}-1} \\ B =& \frac{\mathrm{e}^{c}}{\mathrm{e}^{c}-1}. \end{aligned}$$

So we can integrate the partial fraction instead,

$$
\begin{aligned}
&= \int_{-\infty}^{a} \frac{(w-1)^2}{w(w-1+\mathrm{e}^c)} \frac{1}{w-1}\, dw \\
&= \frac{\mathrm{e}^c}{\mathrm{e}^c-1} \int \frac{1}{w-1+\mathrm{e}^c}\, dw - \frac{1}{\mathrm{e}^c-1} \int \frac{1}{w}\, dw \\
\gamma_K(a,c) &= \frac{1}{\mathrm{e}^c-1} \left[\mathrm{e}^c \ln(w-1+\mathrm{e}^c) - \ln(w)\right]
\end{aligned}
$$

then, plug-in back $w = \mathrm{e}^c + 1$

$$
\begin{aligned}
\gamma_K(a,c) &= \frac{1}{\mathrm{e}^c-1} \left[\mathrm{e}^c \ln(\mathrm{e}^t+\mathrm{e}^c) - \ln(\mathrm{e}^t+1)\right]_{-\infty}^{a} \\
&= \frac{1}{\mathrm{e}^c-1} \left(\mathrm{e}^c \ln(\mathrm{e}^a+\mathrm{e}^c) - \ln(\mathrm{e}^a+1)\right) - \frac{1}{\mathrm{e}^c-1} \left(\mathrm{e}^c \ln(\mathrm{e}^c)\right) \\
&= \frac{\mathrm{e}^c \ln\left(\mathrm{e}^{a-c}+1\right) - \ln\left(\mathrm{e}^a+1\right)}{\mathrm{e}^c-1} \qquad \text{(B.8.22)}
\end{aligned}
$$

when $c = 0$,

$$
\begin{aligned}
\gamma_K(a,c) &= \ln\left(1+\mathrm{e}^{-a}\right) + \frac{a\mathrm{e}^a}{\mathrm{e}^a+1} - \frac{\mathrm{e}^a}{\mathrm{e}^a+1} + \frac{a}{\mathrm{e}^a+1} \\
&= \ln\left(1+\mathrm{e}^{-a}\right) + a - \frac{\mathrm{e}^a}{\mathrm{e}^a+1} \qquad \text{(B.8.23)}
\end{aligned}
$$

□

Derivation B.8.5. *Let K be a Logistic kernel as in Definition B.8.1 with CDF ν_K as in B.8.1. The partial L2-product of Logistic CCDF at two different central points 0 and $c \in \mathbb{R}$ as in Eqn (5.4.4) for the integral boundary $(-\infty, a]$ where $a \in \mathbb{R}$ is*

$$
\xi_K(a,c) = \begin{cases} \ln(\mathrm{e}^a+1) - a - \frac{1}{\mathrm{e}^a+1} & \textit{if} \qquad c = 0 \\ -\frac{\left(\ln\left(\mathrm{e}^{c-a}\right)+1\right)}{\mathrm{e}^c-1} + \frac{\mathrm{e}^c \ln\left(\mathrm{e}^{-a}+1\right)}{\mathrm{e}^c-1} & \textit{otherwise} \end{cases}
$$

Proof. From Lemma 5.4.2, $\xi_K(a,c)$ is the reflection of $\gamma_K(a,c)$ on the y-axis. Hence, $\xi_K(a,c)$ can be computed by $\gamma_K(-a,-c)$ □

B.9 Gaussian Kernel

Definition B.9.1. *Let the Gaussian kernel be defined as*

$$K(u) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{u^2}{2}} \tag{B.9.1}$$

Derivation B.9.1. *Let K be a Gaussian kernel as defined in Definition B.9.1. The cdf of the Gaussian kernel is*

$$\nu_K(t) = \frac{1}{2}\operatorname{erf}\left(\frac{t}{\sqrt{2}}\right) + \frac{1}{2} \tag{B.9.2}$$

Proof. Let $K(u)$ be the Gaussian kernel, to find the CDF we integrate

$$\nu_K(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{u^2}{2}}\, du = \frac{1}{2}\operatorname{erf}\left(\frac{t}{\sqrt{2}}\right) + \frac{1}{2} \tag{B.9.3}$$

where $\mathrm{Erf}(\frac{t}{\sqrt(2)})$ is an error function defined as

$$\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t \mathrm{e}^{-u^2}\, du$$

□

Derivation B.9.2. *Let K be a Gaussian kernel as in Definition B.9.1. The partial L2-product of Gaussian kernel at two different central points 0 and $c \in \mathbb{R}$ for the integration boundary $(-\infty, \infty)$ is*

$$\lambda_K(c) = \frac{1}{2\sqrt{\pi}}\mathrm{e}^{-\frac{c^2}{4}} \tag{B.9.4}$$

Proof. Suppose that we have the Gaussian kernel as in Definition B.9.1. The partial L2-product of Gaussian kernel as in Eqn (5.4.3) for Gaussian kernel is

$$\lambda_K(c) = \int_{-\infty}^{\infty} \frac{1}{2\pi}\mathrm{e}^{-\frac{(u-c)^2}{2}}\mathrm{e}^{-\frac{u^2}{2}}\, du \tag{B.9.5}$$

Then,

$$\lambda_K(a = \infty, c) = \frac{1}{2\sqrt{\pi}}\mathrm{e}^{-\frac{c^2}{4}} \tag{B.9.6}$$

□

Derivation B.9.3. *Let K be a Gaussian kernel as in Definition B.9.1. The partial L2-product of two Gaussian kernels with different central points 0 and $c \in \mathbb{R}$ for the integration boundary $(-\infty, a]$ is*

$$\lambda_K(a,c) = \frac{e^{-\frac{c^2}{4}}}{4\sqrt{\pi}} \left(\operatorname{erf}\left(a - \frac{c}{2}\right) + 1 \right) \tag{B.9.7}$$

Proof. Suppose that we have a Gaussian kernel as in Definition B.9.1. The partial L2-product of the kernel as in Eqn (5.4.3) for Gaussian kernel is

$$\lambda_K(a,c) = \int_{-\infty}^{a} \frac{1}{2\pi} e^{-\frac{(u-c)^2}{2}} e^{-\frac{u^2}{2}}\, du \tag{B.9.8}$$

Then,

$$\lambda_K(a,c) = \frac{e^{-\frac{c^2}{4}}}{4\sqrt{\pi}} \left(\operatorname{erf}\left(a - \frac{c}{2}\right) + 1 \right) \tag{B.9.9}$$

□

Derivation B.9.4. *Let K be a Gaussian Kernel as in Definition B.9.1. The integral of the CDF of Gaussian kernel with central point $c \in \mathbb{R}$ and the integration limit $(-\infty, a]$ where $a \in \mathbb{R}$ is*

$$\zeta_K(a,c) = \frac{(c-a)\left(\operatorname{erf}\left(\frac{c-a}{\sqrt{(2)}}\right) - 1\right)}{2} + \frac{e^{\frac{-(c-a)^2}{2}}}{\sqrt{2\pi}} \tag{B.9.10}$$

Proof. Suppose that K is Gaussian kernel with the cdf ν_K. Then, the integral of the CDF is

$$\zeta_K(a,c) = \int_{-\infty}^{a} \frac{1}{2} \operatorname{erf}\left(\frac{t-c}{\sqrt{2}}\right) + \frac{1}{2}\, dt = \frac{(c-a)\left(\operatorname{erf}\left(\frac{c-a}{\sqrt{(2)}}\right) - 1\right)}{2} + \frac{e^{\frac{-(c-a)^2}{2}}}{\sqrt{2\pi}} \tag{B.9.11}$$

□

Derivation B.9.5. *Let K be a Gaussian Kernel as in Definition B.9.1. The integral of the complimentary CDF of Gaussian kernel with central point $c \in \mathbb{R}$ and the* *integration limit* $[a, \infty)$ *where* $a \in \mathbb{R}$ *is*

$$\eta_K(a, c) = \frac{(c-a)\left(\operatorname{erf}\left(\frac{c-a}{\sqrt{(2)}}\right)+1\right)}{2} + \frac{\mathrm{e}^{\frac{-(c-a)^2}{2}}}{\sqrt{2\pi}} \tag{B.9.12}$$

Proof. Suppose that K is Gaussian kernel with the cdf ν_K. Then, the integral of the CDF is

$$\eta_K(a, c) = \int_{-\infty}^{a} \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{t-c}{\sqrt{2}}\right)\, dt = \frac{(c-a)\left(\operatorname{erf}\left(\frac{c-a}{\sqrt{(2)}}\right)+1\right)}{2} + \frac{\mathrm{e}^{\frac{-(c-a)^2}{2}}}{\sqrt{2\pi}} \tag{B.9.13}$$

□

B.10 Sigmoid kernel

Definition B.10.1. *Let the Sigmoid kernel be defined as*

$$K(u) = \frac{2}{\pi}\frac{1}{e^{u} + e^{-u}} \tag{B.10.1}$$

Derivation B.10.1. *Let K be a Sigmoid kernel as in Def B.10.1. The CDF of a Sigmoid kernel is*

$$\nu_K(t) = \frac{2}{\pi}\arctan(\mathrm{e}^{t}) \tag{B.10.2}$$

Proof. Let K be a Sigmoid kernel as in Definition B.10.1. The CDF is the integration of the kernel function.

$$\nu_K(t) = \int_{-\infty}^{t} \frac{2}{\pi}\frac{1}{\mathrm{e}^{u} + \mathrm{e}^{-u}}du. \tag{B.10.3}$$

Multiply the above with $\frac{\mathrm{e}^{u}}{\mathrm{e}^{u}}$ to get

$$\nu_K(t) =\frac{2}{\pi}\int_{-\infty}^{t} \frac{\mathrm{e}^{u}}{\mathrm{e}^{2u} + 1}\,du. \tag{B.10.4}$$

Make a substitution, $v = \mathrm{e}^{u}$ obtain

$$\nu_K(t) =\frac{2}{\pi}\int \frac{1}{v^2 + 1}\,dv = \frac{2}{\pi}\arctan(v) + C \tag{B.10.5}$$

substitute back $v = \mathrm{e}^{u}$ and put in the limit,

$$\nu_K(t) = \left[\frac{2}{\pi}\arctan(\mathrm{e}^{u})\right]_{-\infty}^{x} = \frac{2}{\pi}\arctan(\mathrm{e}^{t}) \tag{B.10.6}$$

□

Derivation B.10.2. *Let K be a Sigmoid kernel as in Definition B.10.1. The partial L2-product of two Sigmoid kernel at two different starting points 0 and $c \in \mathbb{R}$ when the boundary is $(-\infty, \infty)$ is*

$$\lambda_K(a = \infty, a) = \frac{4c\mathrm{e}^{c}}{\pi^2\left(\mathrm{e}^{2c} - 1\right)} \tag{B.10.7}$$

Proof. Let K be a Sigmoid kernel as in Def B.10.1. The partial L2-product of the

Sigmoid kernel is

$$\begin{aligned}\lambda_K(c) =& \int_{-\infty}^{a=\infty} \frac{2}{\pi} \frac{1}{e^u + e^{-u}} \frac{2}{\pi} \frac{1}{e^{u-c} + e^{-(u-c)}}\, du \\ =& \frac{4}{\pi^{2\frac{-2\mathrm{e}^c}{\pi}}} \int_{-\infty}^{\infty} \frac{1}{\mathrm{e}^{2u-c} + \mathrm{e}^c + \mathrm{e}^{-c} + \mathrm{e}^{-2u+c}}\, du \qquad \text{(B.10.8)} \\ & \qquad \text{(B.10.9)}\end{aligned}$$

Multiply by $\frac{\mathrm{e}^c}{\mathrm{e}^c}$

$$\lambda_K(c) = \frac{4\mathrm{e}^c}{\pi^2} \int_{-\infty}^{\infty} \frac{1}{(\mathrm{e}^u + \mathrm{e}^{2tuc})(\mathrm{e}^u + \mathrm{e}^u)}\, du \qquad \text{(B.10.10)}$$

Multiply by $\frac{\mathrm{e}^{2u}}{\mathrm{e}^{2u}}$,

$$= \frac{4\mathrm{e}^c}{\pi^2} \int_{-\infty}^{\infty} \frac{\mathrm{e}^{2u}}{(\mathrm{e}^{2u} + \mathrm{e}^{2c})(\mathrm{e}^{2u} + 1)}\, du \qquad \text{(B.10.11)}$$

Apply substitution, $w = -2u$, $\frac{dw}{du} = -2$, $du = \frac{-1}{2} dw$

$$= \frac{-2\mathrm{e}^c}{\pi^2} \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-w}}{(\mathrm{e}^{-w} + \mathrm{e}^{2c})(\mathrm{e}^{-w} + 1)}\, dw \qquad \text{(B.10.12)}$$

Multiply by $\frac{\mathrm{e}^{2w}}{\mathrm{e}^{2w}}$

$$= \frac{-2\mathrm{e}^c}{\pi^2} \int_{-\infty}^{\infty} \frac{\mathrm{e}^{w}}{(1 + \mathrm{e}^{w})(\mathrm{e}^{2c+w} + 1)}\, dw \qquad \text{(B.10.13)}$$

Make a substitution, $v = 1 + \mathrm{e}^{2c+w}$ and $\frac{dv}{dw} = \mathrm{e}^{2c+w}$ and $dw = \frac{1}{\mathrm{e}^{2c+w}}\, dv$

$$= \frac{-2\mathrm{e}^c}{\pi^2} \int_{-\infty}^{\infty} \frac{1}{v(v - 1 + \mathrm{e}^{2c})}\, dv \qquad \text{(B.10.14)}$$

Apply partial differential equation to the terms inside the integral, to obtain

$$= \frac{2\mathrm{e}^c}{\pi^2(\mathrm{e}^{2c} - 1)} \left[\ln(\mathrm{e}^{2c-2u} + \mathrm{e}^{2c}) - \ln(1 + \mathrm{e}^{2c-2u})\right]_{-\infty}^{\infty}. \qquad \text{(B.10.15)}$$

Using Approximation of L'Hopital rule, $\lim_{u\to\infty} \ln\left(\frac{\mathrm{e}^{2c-2u}+\mathrm{e}^{2c}}{1+\mathrm{e}^{2c-2x}}\right) \to 2c$, and $\lim_{u\to-\infty} \ln\left(\frac{\mathrm{e}^{2c-2u}+\mathrm{e}^{2c}}{1+\mathrm{e}^{2c-2x}}\right) \to 0$ hence we obtain

$$\lambda_K(a = \infty, c) = \frac{4c\mathrm{e}^c}{\pi^2\left(\mathrm{e}^{2c} - 1\right)}. \qquad \text{(B.10.16)}$$

□

Derivation B.10.3. *Let K be a Sigmoid kernel as in Definition B.10.1. The partial L2-product of two Sigmoid kernel at two different central points 0 and $c \in \mathbb{R}$ when the integration limit is $(-\infty, a]$ where $a \in \mathbb{R}$ is*

$$\lambda_K(a,c) = -\frac{2\mathrm{e}^c\left(\ln\left(\mathrm{e}^{2c}+\mathrm{e}^{2a}\right)-2c-\ln\left(\mathrm{e}^{2a}+1\right)\right)}{\pi^2\left(\mathrm{e}^c-1\right)\left(\mathrm{e}^c+1\right)} \tag{B.10.17}$$

When $c = 0$,

$$\lambda(a,c=0) = \int_{-\infty}^{a} K(u)K(u-c)\,du \tag{B.10.18}$$

$$=\frac{1+\tanh(a)}{\pi^2} \tag{B.10.19}$$

Proof. Suppose that we have Sigmoid kernel as in Definition B.10.1. The partial L2-product of Sigmoid kernel is

$$\lambda_K(a=\infty,c) = \int_{-\infty}^{a=\infty} \frac{2}{\pi}\frac{1}{e^u+e^{-u}}\frac{2}{\pi}\frac{1}{e^{u-c}+e^{-(u-c)}}\,du. \tag{B.10.20}$$

Using the Eqn (B.10.15) and change the limit of the integration from $(-\infty,\infty)$ to $(-\infty,a]$ to obtain,

$$\lambda_K(a,c) =\frac{-2\mathrm{e}^c}{\pi^2(\mathrm{e}^{2c}-1)}\left[\ln(1+\mathrm{e}^{2c-2t})-\ln(\mathrm{e}^{2c-2t}+\mathrm{e}^{2c})\right]_{-\infty}^{a}$$

$$= -\frac{2\mathrm{e}^c\left(\ln\left(\mathrm{e}^{2c}+\mathrm{e}^{2a}\right)-2c-\ln\left(\mathrm{e}^{2a}+1\right)\right)}{\pi^2\left(\mathrm{e}^c-1\right)\left(\mathrm{e}^c+1\right)} \tag{B.10.21}$$

□

B.11 Cosine kernel

Definition B.11.1. *Let a Cosine kernel be defined as*

$$K(u) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)\mathbb{1}(u \in [-1,1]) \tag{B.11.1}$$

Derivation B.11.1. *Let K be a cosine kernel as in Definition B.11.1. The CDF of the Cosine kernel is*

$$\nu_K(t) = \begin{cases} 0 & \textit{if} \quad t \leq -1 \\ \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right) & \textit{if} \quad t \in [-1,1] \\ 1 & \textit{if} \quad t \geq 1 \end{cases}$$

Proof. Let K be a cosine kernel as in Definition B.11.1. The CDF is the integration of the kernel function. There are three cases to considered: (1) $t \leq -1$; (2) $t \in [-1,1]$ (3) $t \geq 1$.

1. Case 1 $t \leq 1$: Under this case, the integration is $-\int_{-\infty}^{-1} K(u)\,du = 0$ because the integration is outside the boundary/limit where the Cosine kernel is defined.
2. Case 2 $t \in [-1,1]$: Under this condition, the boundary of the integration will be $t \in [-1,1]$,

$$\nu_K(t) = \int_{-1}^{t} \frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)\,du = \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right) \tag{B.11.2}$$

Case 3 $t \geq 1$**:** Under this condition, the integration is $\nu_K(t) = 1$. □

Derivation B.11.2. *Let K be a Cosine kernel as in Def B.11.1. The partial L2-product of the kernel at two different central points 0 and $c \in \mathbb{R}$ for the integration boundary is $(-\infty,\infty)$ is*

$$\lambda_K(c) = \begin{cases} -\frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c-2\pi}{2}\right)-\sin\left(\frac{\pi c}{2}\right)+(\pi c-2\pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if} \quad c \geq 0 \\ \frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c+2\pi}{2}\right)-\sin\left(\frac{\pi c}{2}\right)+(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if} \quad c \leq 0 \end{cases}$$

Proof. Suppose we have a Cosine kernel as in Def B.11.1. The partial L2-product of the cosine kernel as in Eqn (5.4.2) is

$$\lambda_K(c) = \frac{\pi^2}{16} \int \cos\left(\frac{\pi}{2}u\right) \cos\left(\frac{\pi}{2}(u-c)\right) \mathbb{1}(u \in [-1,1] \cap [c-1, c+1])\, du. \tag{B.11.3}$$

Firstly, we apply trigonometric rules on λ_K, to show

$$\cos\left(\frac{\pi}{2}u\right) \cos\left(\frac{\pi}{2}(u-c)\right) = \frac{1}{2}\left(\cos\left(\frac{2\pi u - \pi c}{2}\right) + \cos\left(\frac{\pi c}{2}\right)\right) \tag{B.11.4}$$

Then, integrate with limit α and β

$$\frac{\pi^2}{16}\int_\alpha^\beta \frac{1}{2}\left(\cos\left(\frac{2\pi u - \pi c}{2}\right) + \cos\left(\frac{\pi c}{2}\right)\right) du = \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right) u\right] \tag{B.11.5}$$

Using the above equation, we change the limit to $\alpha = -\infty$ and $\beta = \infty$. There are two cases to considered: (1)$c \geq 0$; (2) $c \leq 0$.

1. Case 1 $c \geq 0$: For this case, the limit of the integration is $[c-1, 1]$. So, the limit in Eqn (B.11.5) is changed to $\alpha = c-1$ and $\beta = 1$,

$$\begin{aligned}\lambda_K(c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right) u\right]_{c-1}^{1} \\ =& -\frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c - 2\pi}{2}\right) - \sin\left(\frac{\pi c}{2}\right) + (\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right)\right)\end{aligned} \tag{B.11.6}$$

2. Case 2 $c \leq 0$: For this case, the limit of the integration is $[-1, c+1]$. o, the limit in Eqn (B.11.5) is changed to $\alpha = -1$ and $\beta = c+1$

$$\begin{aligned}\lambda_K(c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right) u\right]_{-1}^{c+1} \\ =& \frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c + 2\pi}{2}\right) - \sin\left(\frac{\pi c}{2}\right) + (\pi c + 2\pi)\cos\left(\frac{\pi c}{2}\right)\right)\end{aligned} \tag{B.11.7}$$

□

Derivation B.11.3. *Let K be a Cosine kernel as in Def B.11.1. The partial L2-product the kernel at two different central points 0 and $c \in \mathbb{R}$ for the integration boundary $[-\infty, a)$ where $a \in \mathbb{R}$ is*

1. For $c \geq 0$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{\pi}{32}\left(-\sin\left(\frac{\pi c - 2\pi a}{2}\right) + \sin\left(\frac{\pi c}{2}\right)\right) - & \\ \frac{\pi}{32}\left((\pi c - \pi a - \pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if } a \in [c-1, 1] \\ \frac{\pi}{32}\left(-\sin\left(\frac{\pi c - 2\pi}{2}\right) + \sin\left(\frac{\pi c}{2}\right)\right) - & \\ \frac{\pi}{32}\left((\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if } a \geq 1 \end{cases}$$

2. For $c \leq 0$

$$\lambda_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq -1 \\ \frac{\pi}{32}\left(-\sin\left(\frac{\pi c - 2\pi a}{2}\right) - \sin\left(\frac{\pi c}{2}\right)\right) + & \\ \frac{\pi}{32}\left((\pi a + \pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if} \quad a \in [-1, c+1] \\ \frac{\pi}{32}\left(\sin\left(\frac{\pi c - 2\pi}{2}\right) - \sin\left(\frac{\pi c}{2}\right)\right) + & \\ \frac{\pi}{32}\left((\pi c + 2\pi)\cos\left(\frac{\pi c}{2}\right)\right) & \textit{if} \quad a \geq c+1 \end{cases}$$

Proof. Suppose that we have a Cosine kernel as in Definition B.11.1. Then, the partial L2-product of two Cosine kernels as in Eqn (5.4.2) is

$$\lambda_K(a,c) = \frac{\pi^2}{16}\int_{-\infty}^{a} \cos\left(\frac{\pi}{2}u\right)\cos\left(\frac{\pi}{2}(u-c)\right)\mathbb{1}(u \in [-1,1] \cap [c-1, c+1])\,du$$

There are three cases to consider: (1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $|c| \geq 0$. For (3), this is outside the intersection of the two Cosine kernels. Resulting $\lambda_K(a,c) = 0$. Hence, the computation only focusses on (1) and (2).

1. **Case 1:** $c \geq 0$ Under this case, $a \in [c-1, 1]$.
   a. Case 1(a) $a \leq -1$:

$$\lambda_K(a,c) = 0 \tag{B.11.8}$$

   b. Case 1(b) $a \in [c-1, 1]$:

$$\begin{aligned} \lambda_K(a,c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right)u\right]_{c-1}^{a} \\ =& \frac{\pi}{32}\left(-\sin\left(\frac{\pi c - 2\pi a}{2}\right) + \sin\left(\frac{\pi c}{2}\right) - (\pi c - \pi a - \pi)\cos\left(\frac{\pi c}{2}\right)\right) \end{aligned} \tag{B.11.9}$$

c. Case 1(c) $a \geq 1$:

$$\begin{aligned}\lambda_K(a,c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right)u\right]_{c-1}^{1} \\ =& \frac{\pi^2}{32\pi}\left(-\sin\left(\frac{\pi c - 2\pi}{2}\right) + \sin\left(\frac{\pi c}{2}\right) - (\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right)\right)\end{aligned} \tag{B.11.10}$$

2. **Case 2:** $c \leq 0$ Under this case, a can be anywhere between $[-1, c+1]$.

a. Case 2(a) $a \leq c - 1$:

$$\lambda_K(a,c) = 0 \tag{B.11.11}$$

b. Case 2(b) $a \in [-1, c+1]$:

$$\begin{aligned}\lambda_K(a,c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right)u\right]_{-1}^{a} \\ =& -\frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c - 2\pi a}{2}\right) + \sin\left(\frac{\pi c}{2}\right) + (-\pi a - \pi)\cos\left(\frac{\pi c}{2}\right)\right)\end{aligned} \tag{B.11.12}$$

c. Case 2(c) $a \geq c + 1$:

$$\begin{aligned}\lambda_K(a,c) =& \frac{\pi^2}{32}\left[\frac{\sin\left(\pi u - \frac{\pi c}{2}\right)}{\pi} + \cos\left(\frac{\pi c}{2}\right)u\right]_{-1}^{c+1} \\ =& \frac{\pi^2}{32\pi}\left(\sin\left(\frac{\pi c + 2\pi}{2}\right) - \sin\left(\frac{\pi c}{2}\right) + (\pi c + 2\pi)\cos\left(\frac{\pi c}{2}\right)\right)\end{aligned} \tag{B.11.13}$$

□

Derivation B.11.4. *Let K be a Cosine kernel as in Def B.11.1 and the CDF is in B.11.1. The partial L2-product of the Cosine CDF with two different central points 0 and $c \in \mathbb{R}$ for the limit $(-\infty, a]$ where $a \in \mathbb{R}$ is*

1. *For $c \in [0,2]$:*

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq c-1 \\ \frac{-4\cos\left(\frac{\pi c - \pi a}{2}\right)+\sin\left(\frac{\pi c - 2\pi a}{2}\right)+3\sin\left(\frac{\pi c}{2}\right)-(\pi c - \pi a - \pi)\cos\left(\frac{\pi c}{2}\right)+}{8\pi} & \\ \frac{-2\pi c - 4\cos\left(\frac{\pi a}{2}\right)+2\pi a+2\pi}{8\pi} & \textit{if } a \in [c-1,1] \\ \frac{6\sin\left(\frac{\pi c}{2}\right)-8\cos\left(\frac{\pi c-\pi a}{2}\right)+(2\pi-\pi c)\cos\left(\frac{\pi c}{2}\right)-2\pi c+4\pi a}{8\pi} & \textit{if } a \in [1,c+1] \\ \frac{6\sin\left(\frac{\pi c}{2}\right)+(2\pi-\pi c)\cos\left(\frac{\pi c}{2}\right)-6\pi c-4\pi+8\pi a}{8\pi} & \textit{if } a \geq c+1 \end{cases}$$

2. *For $c \in [-2,0]$:*

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if } a \leq -1 \\ \frac{-4\cos\left(\frac{\pi c - \pi a}{2}\right)+\sin\left(\frac{\pi c - 2\pi a}{2}\right)-3\sin\left(\frac{\pi c}{2}\right)+(\pi a + \pi)\cos\left(\frac{\pi c}{2}\right)}{8\pi}+ & \\ \frac{2\pi a+2\pi-4\cos\left(\frac{\pi a}{2}\right)}{8\pi} & \textit{if } a \in [-1,c+1] \\ \frac{-6\sin\left(\frac{\pi c}{2}\right)+(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)-8\cos\left(\frac{\pi a}{2}\right)-2\pi c+4\pi a}{8\pi} & \textit{if } a \in [c+1,1] \\ \frac{-6\sin\left(\frac{\pi c}{2}\right)+(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)-2\pi c-4\pi+8\pi a}{8\pi} & \textit{if } a \geq 1 \end{cases}$$

3. *For $c \geq 2$:*

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq c-1 \\ -\frac{2\cos\left(\frac{\pi(c-a)}{2}\right)}{\pi} - c + a + \frac{1}{2} & \textit{if} \quad a \in [c-1,c+1] \\ a-c & \textit{if} \quad a \geq c+1 \end{cases}$$

4. *For $c \leq -2$:*

$$\gamma_K(a,c) = \begin{cases} 0 & \textit{if} \quad a \leq -1 \\ \frac{-\frac{2\cos\left(\frac{\pi a}{2}\right)}{\pi}+a+1}{2} & \textit{if} \quad a \in [-1,1] \\ a & \textit{if} \quad a \geq 1 \end{cases}$$

Proof. Suppose K is a Cosine kernel as in Def B.11.1 with the CDF as in B.11.1. The partial L2-product of Cosine CDF as in Eqn (5.4.3)

$$\gamma_K(a,c) = \int_{-\infty}^{a} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi t - \pi c}{2}\right)+1\right)\, dt \qquad \text{(B.11.14)}$$

There are several cases to be considered in computing $\gamma_K(a,c)$ for Cosine kernel:

(1) $c \in [0,2]$; (2) $c \in [-2,0]$; (3) $c \geq 2$; (4) $c \leq -2$.

1. **Case 1** $c \in [0,2]$**:**
   a. Case 1(a) $a \leq c-1$:

$$\gamma_K(a,c) = 0 \tag{B.11.15}$$

   b. Case 1(b) $a \in [c-1,1]$:

$$\begin{aligned}\gamma_K(a,c) = &\int_{c-1}^{a} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\left(t-c\right)}{2}\right)+1\right)\\ =&\frac{-4\cos\left(\frac{\pi c-\pi a}{2}\right)+\sin\left(\frac{\pi c-2\pi a}{2}\right)+4\cos\left(\frac{\pi c-\pi}{2}\right)}{8\pi}+\\ &\frac{\sin\left(\frac{\pi c-2\pi}{2}\right)-(\pi c-\pi a-\pi)\cos\left(\frac{\pi c}{2}\right)}{8\pi}+\\ &\frac{\left(-2\pi c-4\cos\left(\frac{\pi a}{2}\right)\right)+2\pi a+2\pi}{8\pi}\end{aligned} \tag{B.11.16}$$

Check: For $a = c-1$

$$\gamma_K(a,c) = \int_{c-1}^{c-1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\left(t-c\right)}{2}\right)+1\right) = 0 \tag{B.11.17}$$

Check: When $a = 1$

$$\begin{aligned}\gamma_K(a,c) = &\int_{c-1}^{1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\left(t-c\right)}{2}\right)+1\right)\\ =&\frac{4\cos\left(\frac{\pi c-\pi}{2}\right)+\sin\left(\frac{\pi c-2\pi}{2}\right)-5\sin\left(\frac{\pi c}{2}\right)}{8\pi}+\\ &\frac{(2\pi-\pi c)\cos\left(\frac{\pi c}{2}\right)-2\pi c+4\pi}{8\pi}\end{aligned} \tag{B.11.18}$$

   c. Case 1(c) $a \in [1,c+1]$:

$$\begin{aligned}\gamma_K(a,c) = &\int_{c-1}^{1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\left(t-c\right)}{2}\right)+1\right)+\\ &\int_{1}^{a} \frac{1}{2}\left(\sin\left(\frac{\pi\left(t-c\right)}{2}\right)+1\right)\,dt\\ =&\frac{\sin\left(\frac{\pi c-2\pi}{2}\right)+12\cos\left(\frac{\pi c-\pi}{2}\right)-8\cos\left(\frac{\pi c-\pi a}{2}\right)-5\sin\left(\frac{\pi c}{2}\right)}{8\pi}+\\ &\frac{(2\pi-\pi c)\cos\left(\frac{\pi c}{2}\right)-2\pi c+4\pi a}{8\pi}\end{aligned} \tag{B.11.19}$$

Check: When $a = c + 1$

$$\begin{aligned}\gamma_K(a,c) =& \frac{12\cos\left(\frac{\pi c-\pi}{2}\right) + \sin\left(\frac{\pi c-2\pi}{2}\right) - 5\sin\left(\frac{\pi c}{2}\right)}{8\pi} + \\ & \frac{(2\pi - \pi c)\cos\left(\frac{\pi c}{2}\right) + 2\pi c + 4\pi}{8\pi}\end{aligned} \tag{B.11.20}$$

d. Case 1(d) $a \geq c + 1$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{c-1}^{1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right) + 1\right)\frac{1}{2}\left(\sin\left(\frac{\pi (t-c)}{2}\right) + 1\right)\, dt + \\ & \int_{1}^{c+1} \frac{1}{2}\left(\sin\left(\frac{\pi (t-c)}{2}\right) + 1\right)\, dt + \int_{c+1}^{a} 1\, dt \\ =& \frac{12\cos\left(\frac{\pi c-\pi}{2}\right) + \sin\left(\frac{\pi c-2\pi}{2}\right) - 5\sin\left(\frac{\pi c}{2}\right)}{8\pi} + \\ & \frac{(2\pi - \pi c)\cos\left(\frac{\pi c}{2}\right) - 6\pi c - 4\pi + 8\pi a}{8\pi}\end{aligned} \tag{B.11.21}$$

2. **For $c \in [-2, 0]$:**

a. Case 2(a) $a \leq -1$:

$$\gamma_K(a,c) = 0 \tag{B.11.22}$$

b. Case 2(b) $a \in [-1, c + 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{a} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right) + 1\right)\frac{1}{2}\left(\sin\left(\frac{\pi (t-c)}{2}\right) + 1\right)\, dt \\ =& \frac{-4\cos\left(\frac{\pi c-\pi a}{2}\right) + \sin\left(\frac{\pi c-2\pi a}{2}\right) - 3\sin\left(\frac{\pi c}{2}\right) +}{8\pi} \\ & \frac{(\pi a + \pi)\cos\left(\frac{\pi c}{2}\right) - 4\cos\left(\frac{\pi a}{2}\right) + 2\pi a + 2\pi}{8\pi}\end{aligned} \tag{B.11.23}$$

Check: When $a = c + 1$

$$\begin{aligned}\gamma_K(a,c) =& \frac{-\sin\left(\frac{\pi c+2\pi}{2}\right) - 4\cos\left(\frac{\pi c+\pi}{2}\right) - 3\sin\left(\frac{\pi c}{2}\right) + (\pi c + 2\pi)\cos\left(\frac{\pi c}{2}\right)}{8\pi} + \\ & \frac{2\pi c + 4\pi}{8\pi}\end{aligned} \tag{B.11.24}$$

c. Case 2(c) $a \in [c+1, 1]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\,(t-c)}{2}\right)+1\right)\,dt+\\ & \int_{c+1}^{a} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\,dt\\ =&\frac{-\sin\left(\frac{\pi c+2\pi}{2}\right)+4\cos\left(\frac{\pi c+\pi}{2}\right)-3\sin\left(\frac{\pi c}{2}\right)}{8\pi}+\\ & \frac{(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)-8\cos\left(\frac{\pi a}{2}\right)-2\pi c+4\pi a}{8\pi}\end{aligned} \tag{B.11.25}$$

Check: When $a = 1$

$$\begin{aligned}\gamma_K(a,c) =&\frac{-\sin\left(\frac{\pi c+2\pi}{2}\right)+4\cos\left(\frac{\pi c+\pi}{2}\right)-3\sin\left(\frac{\pi c}{2}\right)}{8\pi}+\\ & \frac{(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)-2\pi c+4\pi}{8\pi}\end{aligned} \tag{B.11.26}$$

d. Case 2(d) $a \in [1, \infty]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-1}^{c+1} \frac{1}{2}\left(\sin\left(\frac{\pi t}{2}\right)+1\right)\frac{1}{2}\left(\sin\left(\frac{\pi\,(t-c)}{2}\right)+1\right)\,dt+\\ =&\frac{-\sin\left(\frac{\pi c+2\pi}{2}\right)+4\cos\left(\frac{\pi c+\pi}{2}\right)-3\sin\left(\frac{\pi c}{2}\right)}{8\pi}+\\ & \frac{(\pi c+2\pi)\cos\left(\frac{\pi c}{2}\right)-2\pi c-4\pi+8\pi a}{8\pi}\end{aligned} \tag{B.11.27}$$

3. **Case 3** $c \geq 2$**:** Under this condition, the intersection of the two cdf's ν_K with central points 0 and $c \in \mathbb{R}$ happens:

a. Case 3(a) $a \in [c-1, c+1]$:

$$\gamma_K(a,c) = \int_{c-1}^{a}\left(\sin\left(\frac{\pi\,(t-c)}{2}\right)+1\right)\,dt = -\frac{2\cos\left(\frac{\pi(c-a)}{2}\right)}{\pi} - c + a + \frac{1}{2} \tag{B.11.28}$$

b. Case 3(b) $a \geq c+1$:

$$\gamma_K(a,c) = \int_{c-1}^{c+1}\left(\sin\left(\frac{\pi\,(t-c)}{2}\right)+1\right)\,dt + \int_{c+1}^{a} 1\,dt = a-c \tag{B.11.29}$$

4. **Case 4** $c \leq -2$**:** Under this condition, the intersection of the two cdf's ν_K with

central points 0 and $c \in \mathbb{R}$ happens:

a. Case 4(a) $a \in [-1, 1]$:

$$\gamma_K(a, c) = \int_{-1}^{a} \left(\sin\left(\frac{\pi t}{2}\right) + 1 \right) \, dt = \frac{-\frac{2\cos\left(\frac{\pi a}{2}\right)}{\pi} + a + 1}{2} \tag{B.11.30}$$

b. Case 4(b) $a \geq 1$:

$$\gamma_K(a, c) = \int_{-1}^{1} \left(\sin\left(\frac{\pi t}{2}\right) + 1 \right) \, dt + \int_{1}^{a} 1 \, dt = a \tag{B.11.31}$$

□

Derivation B.11.5. *Let K be Cosine kernel as in Def B.11.1 with CDF as in B.11.4. The integration of $(1 - \nu_K)$ at two different central points 0 and $c \in \mathbb{R}$ as in Eqn (5.4.11) is*

1. ***For** $c \in [0, 2]$**:***

$$\xi_K(a, c) = \begin{cases} \frac{6\sin\left(\frac{\pi c}{2}\right) - (\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right) - 2\pi c - 8\pi a}{8\pi} & \textit{if } a \leq -1 \\ \frac{6\sin\left(\frac{\pi c}{2}\right) - (\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right) - 8\cos\left(\frac{\pi a}{2}\right) + 2\pi c - 4\pi a}{8\pi} & \textit{if } a \in [-1, c-1] \\ \frac{-4\cos\left(\frac{\pi c - \pi a}{2}\right) - \sin\left(\frac{\pi c - 2\pi a}{2}\right) + 3\sin\left(\frac{\pi c}{2}\right)}{8\pi} - & \\ \frac{(\pi a - \pi)\cos\left(\frac{\pi c}{2}\right) + 4\cos\left(\frac{\pi a}{2}\right) + 2\pi a - 2\pi}{8\pi} & \textit{if } a \in [c-1, 1] \\ 0 & \textit{if } a \geq 1 \end{cases}$$

2. ***For** $c \in [-2, 0]$**:***

$$\xi_K(a, c) = \begin{cases} \frac{-6\sin\left(\frac{\pi c}{2}\right) - (-\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right) + 6\pi c - 4\pi - 8\pi a}{8\pi} & \textit{if } a \leq c-1 \\ \frac{-6\sin\left(\frac{\pi c}{2}\right) - 8\cos\left(\frac{\pi c - \pi a}{2}\right) - (-\pi c - 2\pi)\cos\left(\frac{\pi c}{2}\right) + 2\pi c - 4\pi a}{8\pi} & \textit{if } a \in [c-1, -1] \\ \frac{-4\cos\left(\frac{\pi c - \pi a}{2}\right) - \sin\left(\frac{\pi c - 2\pi a}{2}\right) - 3\sin\left(\frac{\pi c}{2}\right)}{8\pi} - & \\ \frac{(-\pi c + \pi a - \pi)\cos\left(\frac{\pi c}{2}\right) - 2\pi c + 4\cos\left(\frac{\pi a}{2}\right) + 2\pi a - 2\pi}{8\pi} & \textit{if } a \in [-1, c+1] \\ 0 & \textit{if } a \geq c+1 \end{cases}$$

3. ***For** $c \geq 2$**:***

$$\gamma_K(a, c) = \begin{cases} -a & \textit{if} \quad a \leq -1 \\ \frac{\frac{2\cos\left(\frac{\pi a}{2}\right)}{\pi} - a + 1}{2} & \textit{if} \quad a \in [-1, 1] \\ 0 & \textit{if} \quad a \geq 1 \end{cases}$$

4. ***For*** $c \leq -2$***:***

$$\gamma_K(a,c) = \begin{cases} c - a & \textit{if} \quad a \leq c-1 \\ -\frac{2\cos\left(\frac{\pi(a-c)}{2}\right)}{\pi} + c - a + \frac{1}{2} & \textit{if} \quad a \in [c-1, c+1] \\ 0 & \textit{if} \quad a \geq c+1 \end{cases}$$

Proof. From Lemma 5.4.2, $\xi_K(a,c)$ is the reflection of $\gamma_K(a,c)$ on the y-axis. Hence, we can compute $\xi_K(a,c)$ by taking $\gamma_K(-a,-c)$.

□

B.12 Silverman's kernel

Definition B.12.1. *Let the Silverman's kernel be defined as*

$$K(u) = \frac{1}{2}\mathrm{e}^{\frac{-|u|}{\sqrt{2}}} \sin\left(\frac{|u|}{\sqrt{2}} + \frac{\pi}{4}\right) \tag{B.12.1}$$

such that

$$K(u) = \begin{cases} \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) & \textit{for} \qquad u \geq 0 \\ \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) & \textit{for} \qquad u \leq 0 \end{cases}$$

Derivation B.12.1. *Let K be a Silverman's kernel as in Def B.12.1. The CDF of Silverman's kernel is*

$$\nu_K(t) = \begin{cases} \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) & \textit{if} \qquad t \leq 0 \\ 1 - \frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) & \textit{if} \qquad t \geq 0 \end{cases}$$

Proof. Let K be a Silverman's kernel in Def B.12.1. The CDF is the integration of the kernel function K. There are two cases to consider: (1) $t \leq 0$; (2) $t \geq 0$.

1. Case 1 $t \leq 0$:

$$\nu_K(t) = \int_{-\infty}^{t} \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(-\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right)\, dt = \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \tag{B.12.2}$$

2. Case 2 $t \geq 0$

$$\begin{aligned}\nu_K(t) =& \int_{-\infty}^{0} \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(-\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right)\, dt + \int_{0}^{t} \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right)\, dt \\ =& 1 - \frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \end{aligned} \tag{B.12.3}$$

□

Derivation B.12.2. *Let K be a Silverman's kernel as in Defi B.12.1.The L2-norm of two Silverman kernels at two different central points 0 and $c \in \mathbb{R}$ is for the integration boundary is $(-\infty, \infty)$ is*

$$\lambda_K(c) = \begin{cases} \frac{\mathrm{e}^{-\frac{c}{\sqrt{2}}}\left(3\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{2^{\frac{7}{2}}} + \frac{c\mathrm{e}^{-\frac{c}{\sqrt{2}}}\sin\left(\frac{c}{\sqrt{2}}\right)}{8} & \textit{for} \qquad c \geq 0 \\ \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-3\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{2^{\frac{7}{2}}} + \frac{c\mathrm{e}^{\frac{c}{\sqrt{2}}}\sin\left(\frac{c}{\sqrt{2}}\right)}{8} & \textit{for} \qquad c \leq 0 \end{cases}$$

Proof. Suppose K is a Silverman kernel as in Def B.12.1. The partial L2-product of the Silverman kernel as in Eqn (5.4.2) is There are 2 cases that need to be considered.

1. Case 1 $c \geq 0$:

$$\begin{aligned}\lambda_K(c) = &\int_{-\infty}^{0} \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u-c}{\sqrt{2}}} \sin\left(\frac{-(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) du + \\ &\int_{0}^{c} \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u-c}{\sqrt{2}}} \sin\left(\frac{-(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) du + \\ &\int_{c}^{\infty} \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{-(u-c)}{\sqrt{2}}} \sin\left(\frac{(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) du \\ = &\frac{\mathrm{e}^{-\frac{c}{\sqrt{2}}}\left(3\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{2^{\frac{7}{2}}} + \frac{c\mathrm{e}^{-\frac{c}{\sqrt{2}}}\sin\left(\frac{c}{\sqrt{2}}\right)}{8} \end{aligned} \tag{B.12.4}$$

2. Case 2 $c \leq 0$:

$$\begin{aligned}\lambda_K(c) = &\int_{-\infty}^{c} \frac{1}{2}\mathrm{e}^{\frac{u-c}{\sqrt{2}}} \sin\left(-\frac{u-c}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(-\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) du + \\ &\int_{c}^{0} \frac{1}{2}\mathrm{e}^{-\frac{u-c}{\sqrt{2}}} \sin\left(\frac{u-c}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(-\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) du + \\ &\int_{0}^{\infty} \frac{1}{2}\mathrm{e}^{-\frac{u-c}{\sqrt{2}}} \sin\left(\frac{u-c}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(-\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) du \\ = &\frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-3\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{2^{\frac{7}{2}}} + \frac{c\mathrm{e}^{\frac{c}{\sqrt{2}}}\sin\left(\frac{c}{\sqrt{2}}\right)}{8} \end{aligned} \tag{B.12.5}$$

□

Derivation B.12.3. *Let K be a Silverman's kernel as in Def B.12.1. The partial L2-product of two Silverman's kernel at two different central points 0 and $c \in \mathbb{R}$ from $-\infty$ to $a \in \mathbb{R}$ is*

1. Case 1 $c \geq 0$:

$$\lambda_K(a,c) = \begin{cases} \frac{\mathrm{e}^{-\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{if} \quad a \leq 0 \\ \frac{\mathrm{e}^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} - \\ \frac{\mathrm{e}^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\left(-\sqrt{2}a-1\right)\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}} & \textit{if} \quad a \in [0,c] \\ \frac{\exp^{\frac{-c}{\sqrt{2}}}\left((3+\sqrt{2}c)\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}} + \\ \frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)-\cos\left(\frac{c-2a}{\sqrt{2}}\right)-2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{if} \quad a \in [c,\infty] \end{cases}$$

2. *Case 1 $c \le 0$:*

$$\lambda_K(a,c) = \begin{cases} \frac{e^{\frac{2a-c}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{if} \quad a \le 0 \\ \frac{e^{\frac{c}{\sqrt{2}}}\left((-2+\sqrt{2}c-\sqrt{2}a)\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)-\sin\left(\frac{c-2a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{if} \quad a \in [c,0] \\ \frac{e^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)-\cos\left(\frac{c-2a}{\sqrt{2}}\right)-2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + \\ \frac{e^{\frac{c}{\sqrt{2}}}\left(3\cos\left(\frac{c}{\sqrt{2}}\right)-(3-\sqrt{2}c)\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}} & \textit{if} \quad a \in [0,\infty) \end{cases}$$

Proof. Suppose K is a Silverman kernel as in Definition B.12.1. The partial L2-product of the Silverman kernel as in Eqn (5.4.2) is There are two cases to be considered: (1) $c \ge 0$; (2) $c \le 0$.

1. **Case 1** $c \ge 0$**:** Under this condition, we need to consider the sub-cases below.

a. Case 1(a) $a \le 0$:

$$\begin{aligned} \lambda_K(a,c) =& \int_{-\infty}^{a} \frac{1}{2} e^{\frac{u}{\sqrt{2}}} \sin\left(\frac{\pi}{4} - \frac{u}{\sqrt{2}}\right) \frac{1}{2} e^{\frac{u-c}{\sqrt{2}}} \sin\left(\frac{\pi}{4} - \frac{u-c}{\sqrt{2}}\right) du \\ =& \frac{1}{8} e^{-\frac{c}{\sqrt{2}}} \int_{-\infty}^{a} e^{u\sqrt{2}} \left(\cos\left(\frac{c}{\sqrt{2}}\right) + \sin\left(\frac{c-2u}{\sqrt{2}}\right)\right) du \end{aligned}$$

Apply integration by part

$$\begin{aligned} &\int_{-\infty}^{a} e^{u\sqrt{2}} \left(\cos\left(\frac{c}{\sqrt{2}}\right) + \sin\left(\frac{c-2u}{\sqrt{2}}\right)\right) du \\ =& \left[\frac{e^{u\sqrt{2}}}{\sqrt{2}} \left(\cos\left(\frac{c}{\sqrt{2}}\right) + \sin\left(\frac{c-2u}{\sqrt{2}}\right)\right)\right]_{-\infty}^{a} + \int_{-\infty}^{a} e^{u\sqrt{2}} \cos\left(\frac{c-2u}{\sqrt{2}}\right) du \end{aligned}$$

Apply again integration by parts to the rhs of the above, to obtain

$$= \left[\frac{e^{u\sqrt{2}}}{2\sqrt{2}} \cos\left(\frac{c-2u}{\sqrt{2}}\right)\right]_{-\infty}^{a} - \left[\frac{e^{u\sqrt{2}}}{2\sqrt{2}} \sin\left(\frac{c-2u}{\sqrt{2}}\right)\right]_{-\infty}^{a}$$

so the integration of $K(u)K(u-c)$ is

$$\begin{aligned} =& \frac{1}{8} e^{-\frac{c}{\sqrt{2}}} \int_{-\infty}^{a} e^{u\sqrt{2}} \left(\cos\left(\frac{c}{\sqrt{2}}\right) + \sin\left(\frac{c-2u}{\sqrt{2}}\right)\right) du \\ =& \frac{e^{-\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \end{aligned} \tag{B.12.6}$$

Check: $a = 0$,

$$\lambda_K(a,c) = \frac{e^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.7}$$

b. Case 1(b) $a \in [0, c]$:

$$\begin{aligned}\lambda_K(a,c) = &\int_{-\infty}^{0} \frac{1}{2}e^{\frac{u}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u}{\sqrt{2}}\right)\frac{1}{2}e^{\frac{u-c}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u-c}{\sqrt{2}}\right)\,du+\\ &\int_{0}^{a} \frac{1}{2}e^{\frac{-u}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u}{\sqrt{2}}\right)\frac{1}{2}e^{\frac{-(u-c)}{\sqrt{2}}}\sin\left(\frac{\pi}{4} + \frac{u-c}{\sqrt{2}}\right)\,du\\ =&\frac{e^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} -\\ &\frac{e^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \left(-\sqrt{2}a - 1\right)\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}}\end{aligned} \tag{B.12.8}$$

Check: When $a = 0$

$$\lambda_K(a,c) = \frac{e^{-\frac{c}{\sqrt{2}}}\left(\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.9}$$

Check: When $a = c$,

$$\lambda_K(a,c) = \frac{e^{-\frac{c}{\sqrt{2}}}\left(\left(2^{\frac{3}{2}}c + 5\right)\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.10}$$

c. Case 1(c) $a \in [c, \infty)$:

$$\begin{aligned}\lambda_K(a,c) = &\int_{-\infty}^{0} \frac{1}{2}e^{\frac{u}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u}{\sqrt{2}}\right)\frac{1}{2}e^{\frac{u-c}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u-c}{\sqrt{2}}\right)\,du\\ &\int_{0}^{c} \frac{1}{2}e^{\frac{-u}{\sqrt{2}}}\sin\left(\frac{\pi}{4} - \frac{u}{\sqrt{2}}\right)\frac{1}{2}e^{\frac{u-c}{\sqrt{2}}}\sin\left(\frac{\pi}{4} + \frac{u-c}{\sqrt{2}}\right)\,du+\\ &\int_{c}^{a} \frac{1}{2}e^{\frac{u}{\sqrt{2}}}\sin\left(\frac{\pi}{4} + \frac{u}{\sqrt{2}}\right)\frac{1}{2}e^{\frac{u-c}{\sqrt{2}}}\sin\left(\frac{\pi}{4} + \frac{u-c}{\sqrt{2}}\right)\,du\\ =&\frac{e^{\frac{-c}{\sqrt{2}}}\left((3 + \sqrt{2}c)\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}} +\\ &\frac{e^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) - \cos\left(\frac{c-2a}{\sqrt{2}}\right) - 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}\end{aligned} \tag{B.12.11}$$

Check: When $a = \infty$

$$\lambda_K(a,c) = \mathrm{e}^{-\frac{c}{\sqrt{2}}} \left(\frac{3\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)}{8\sqrt{2}} + \frac{c\sin\left(\frac{c}{\sqrt{2}}\right)}{8} \right) \tag{B.12.12}$$

2. **Case 2** $c \leq 0$**:** Under this condition, we need to consider the sub-cases below.

a. Case 2(a) $a \leq c$:

$$\begin{aligned}\lambda_K(a,c) &= \int_{-\infty}^{a} \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) \\ &= \frac{\mathrm{e}^{\frac{2a-c}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}\end{aligned}$$

Check: When $a = c$

$$\lambda_K(a,c) = \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.13}$$

b. Case 2(b) $a \in [c, 0]$:

$$\begin{aligned}\lambda_K(a,c) &= \int_{-\infty}^{0} \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) + \\ &\quad \int_{0}^{a} \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2}\mathrm{e}^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) du \\ &= -\frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(2\sin\left(\frac{c-2a}{\sqrt{2}}\right) - \left(2^{\frac{3}{2}}(c-a) - 3\right)\sin\left(\frac{c}{\sqrt{2}}\right) - 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{2^{\frac{9}{2}}}\end{aligned} \tag{B.12.14}$$

Check: When $a = 0$

$$\lambda_K(a,c) = \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-5\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right) + 2\sqrt{2}c\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.15}$$

c. Case 2(c) $a \in [0, \infty)$:

$$\begin{aligned}
\lambda_K(a,c) =& \int_{-\infty}^{c} \frac{1}{2} e^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2} e^{\frac{u-c}{\sqrt{2}}} \sin\left(\frac{-(u-c)}{\sqrt{2}} + \frac{\pi}{4}\right) + \\
& \int_{c}^{0} \frac{1}{2} e^{\frac{u}{\sqrt{2}}} \sin\left(\frac{-u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2} e^{\frac{-(u-c)}{\sqrt{2}}} \sin\left(\frac{u-c}{\sqrt{2}} + \frac{\pi}{4}\right) du \\
& \int_{0}^{a} \frac{1}{2} e^{\frac{-u}{\sqrt{2}}} \sin\left(\frac{u}{\sqrt{2}} + \frac{\pi}{4}\right) \frac{1}{2} e^{\frac{-(u-c)}{\sqrt{2}}} \sin\left(\frac{u-c}{\sqrt{2}} + \frac{\pi}{4}\right) du \\
=& \frac{e^{\frac{c-2a}{\sqrt{2}}} \left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) - \cos\left(\frac{c-2a}{\sqrt{2}}\right) - 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + \\
& \frac{e^{\frac{c}{\sqrt{2}}} \left(3\cos\left(\frac{c}{\sqrt{2}}\right) - (3 - \sqrt{2}c)\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}}
\end{aligned} \tag{B.12.16}$$

Check: When $a = \infty$

$$\lambda_K(a,c) = \frac{e^{\frac{c}{\sqrt{2}}} \left(-3\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right) + \sqrt{2}c\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{8\sqrt{2}} \tag{B.12.17}$$

□

Derivation B.12.4. *Let K be a Silverman kernel as in Def B.12.1. The partial L2-product of two CDF of Silverman kernels at two different centre points 0 and $c \in \mathbb{R}$ from $-\infty$ to $a \in \mathbb{R}$ is*

1. For $c \geq 0$

$$\gamma_K(a,c) = \begin{cases}
\frac{e^{\frac{2a-c}{\sqrt{2}}} \left(-\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for } a \in (-\infty, 0] \\
\frac{8e^{\frac{-c+a}{\sqrt{2}}} \left(-\sin\left(\frac{c-a}{\sqrt{2}}\right) + \cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + & \\
\frac{e^{\frac{-c}{\sqrt{2}}} \left(2\sin\left(\frac{c-2a}{\sqrt{2}}\right) + 5\sin\left(\frac{c}{\sqrt{2}}\right) - \left(2\sqrt{2}a + 5\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for } a \in [0, c] \\
\frac{8e^{\frac{c-a}{\sqrt{2}}} \left(\sin\left(\frac{c-a}{\sqrt{2}}\right) + \cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} - & \\
\frac{e^{\frac{c-2a}{\sqrt{2}}} \left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} - & \\
\frac{8e^{\frac{-a}{\sqrt{2}}} \left((\sin\left(\frac{a}{\sqrt{2}}\right) - \cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + & \\
\frac{e^{\frac{-c}{\sqrt{2}}} \left(10\sin\left(\frac{c}{\sqrt{2}}\right) - (10 + 2\sqrt{2}c)\cos\left(\frac{c}{\sqrt{2}}\right)\right) - 16\sqrt{2}(c-a)}{16\sqrt{2}} & \textit{for } a \in [c, \infty)
\end{cases}$$

2. *For* $c \leq 0$

$$\gamma_K(a,c) = \begin{cases} \frac{\mathrm{e}^{\frac{2a-c}{\sqrt{2}}}\left(-\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for } a \in [-\infty, c) \\ \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-5\sin\left(\frac{c}{\sqrt{2}}\right)+\left(\sqrt{2}(c-a)-5\right)\cos\left(\frac{c}{\sqrt{2}}\right)+2\sin\left(\frac{c-2a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+ & \\ \frac{8\mathrm{e}^{\frac{a}{\sqrt{2}}}\left(\sin\left(\frac{a}{\sqrt{2}}\right)+\cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for } a \in [c, 0] \\ \frac{8\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-a}{\sqrt{2}}\right)+\cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}- & \\ \frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}- & \\ \frac{8\mathrm{e}^{-\frac{a}{\sqrt{2}}}\left(\sin\left(\frac{a}{\sqrt{2}}\right)-\cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+ & \\ \frac{16\sqrt{2}a-\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(10\sin\left(\frac{c}{\sqrt{2}}\right)-\left(2\sqrt{2}c-10\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)-8}{16\sqrt{2}} & \textit{for } a \in [0, \infty) \end{cases}$$

Proof. Let K be a Silverman kernel with CDF as in B.12.1. There are several cases needed to be considered in computing $\gamma_K(a,c)$ for Silverman kernel. For $c \geq 0$ and $c \leq 0$.

1. **Case 1:** $c \geq 0$: Under this condition, we consider the three sub-cases below.
   a. Case 1(a) $a \in (-\infty, 0]$:

$$\begin{aligned} \gamma_K(a,c) &= \int_{-\infty}^{a} \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}}\cos\left(\frac{t}{\sqrt{2}}\right)\frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}}\cos\left(\frac{t-c}{\sqrt{2}}\right)\,dt \\ &= \frac{\mathrm{e}^{-\frac{c-2a}{\sqrt{2}}}\left(-\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \end{aligned} \tag{B.12.18}$$

Check: When $a = 0$

$$\gamma_K(a,c) = \frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(-\sin\left(\frac{c}{\sqrt{2}}\right)+3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.19}$$

b. Case 1(b) $a \in [0, c]$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-\infty}^{0} \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\, dt\\ & \int_{0}^{a} \left(1 - \frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right)\right) \left(\frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\right)\, dt\\ =& \frac{8\mathrm{e}^{\frac{-c+a}{\sqrt{2}}}\left(-\sin\left(\frac{c-a}{\sqrt{2}}\right) + \cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} +\\ & \frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(2\sin\left(\frac{c-2a}{\sqrt{2}}\right) + 5\sin\left(\frac{c}{\sqrt{2}}\right) - \left(2\sqrt{2}a + 5\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}\end{aligned} \tag{B.12.20}$$

Check: When $a = c$,

$$\gamma_K(a,c) = \frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(3\sin\left(\frac{c}{\sqrt{2}}\right) - (5 + 2\sqrt{2}c)\cos\left(\frac{c}{\sqrt{2}}\right)\right) + 8}{16\sqrt{2}} \tag{B.12.21}$$

c. Case 1(c) $a \in [c, \infty)$:

$$\begin{aligned}\gamma_K(a,c) =& \int_{-\infty}^{0} \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\, dt\\ & \int_{0}^{c} \left(1 - \frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right)\right) \left(\frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\right)\, dt+\\ & \int_{c}^{a} \left(1 - \frac{1}{2}\mathrm{e}^{\frac{-(t-c)}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\right) \left(1 - \frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right)\right)\, dt\\ =& \frac{8\mathrm{e}^{\frac{c-a}{\sqrt{2}}}\left(\sin\left(\frac{c-a}{\sqrt{2}}\right) + \cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} -\\ & \frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} -\\ & \frac{8\mathrm{e}^{\frac{-a}{\sqrt{2}}}\left((\sin\left(\frac{a}{\sqrt{2}}\right) - \cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} +\\ & \frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(10\sin\left(\frac{c}{\sqrt{2}}\right) - (10 + 2\sqrt{2}c)\cos\left(\frac{c}{\sqrt{2}}\right)\right) - 16\sqrt{2}(c-a)}{16\sqrt{2}}\end{aligned} \tag{B.12.22}$$

2. **Case 2:** $c \leq 0$**:**Under this condition, we consider the three sub-cases below.

a. Case 2(a) $a \leq c$:

$$\gamma_K(a,c) = \int_{-\infty}^{a} \frac{1}{2} e^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \frac{1}{2} e^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right) dt \tag{B.12.23}$$

$$= \frac{e^{\frac{2a-c}{\sqrt{2}}} \left(-\sin\left(\frac{c-2a}{\sqrt{2}}\right) + \cos\left(\frac{c-2a}{\sqrt{2}}\right) + 2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.24}$$

Check: When $a = c$

$$\gamma_K(a,c) = \frac{e^{\frac{c}{\sqrt{2}}} \left(\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}$$

b. Case 2(b) $a \in [c, 0]$:

$$\gamma_K(a,c) = \int_{-\infty}^{c} \frac{1}{2} e^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right) \frac{1}{2} e^{\frac{t-c}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right) dt +$$
$$\int_{c}^{a} \left(\frac{1}{2} e^{\frac{t}{\sqrt{2}}} \cos\left(\frac{t}{\sqrt{2}}\right)\right) \left(1 - \frac{1}{2} e^{\frac{-(t-c)}{\sqrt{2}}} \cos\left(\frac{t-c}{\sqrt{2}}\right)\right) dt$$
$$= \frac{e^{\frac{c}{\sqrt{2}}} \left(-5\sin\left(\frac{c}{\sqrt{2}}\right) + \left(\sqrt{2}(c-a) - 5\right)\cos\left(\frac{c}{\sqrt{2}}\right) + 2\sin\left(\frac{c-2a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} +$$
$$\frac{8e^{\frac{a}{\sqrt{2}}} \left(\sin\left(\frac{a}{\sqrt{2}}\right) + \cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.25}$$

Check: When $a = c$,

$$\gamma_K(a,c) = \frac{e^{\frac{c}{\sqrt{2}}} \left(\sin\left(\frac{c}{\sqrt{2}}\right) + 3\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.26}$$

Check: When $a = 0$,

$$\gamma_K(a,c) = \frac{e^{\frac{c}{\sqrt{2}}} \left((2\sqrt{2}c - 5)\cos\left(\frac{c}{\sqrt{2}}\right) - 3\sin\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} \tag{B.12.27}$$

c. Case 2(c) $a \geq 0$:

$$
\begin{aligned}
\gamma_K(a,c) = &\int_{-\infty}^{a} \frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}}\cos\left(\frac{t}{\sqrt{2}}\right)\frac{1}{2}\mathrm{e}^{\frac{t-c}{\sqrt{2}}}\cos\left(\frac{t-c}{\sqrt{2}}\right)\,dt+\\
&\int_{c}^{0}\left(\frac{1}{2}\mathrm{e}^{\frac{t}{\sqrt{2}}}\cos\left(\frac{t}{\sqrt{2}}\right)\right)\left(1-\frac{1}{2}\mathrm{e}^{\frac{-(t-c)}{\sqrt{2}}}\cos\left(\frac{t-c}{\sqrt{2}}\right)\right)\,dt+\\
&\int_{0}^{a}\left(1-\frac{1}{2}\mathrm{e}^{\frac{-t}{\sqrt{2}}}\cos\left(\frac{t}{\sqrt{2}}\right)\right)\left(1-\frac{1}{2}\mathrm{e}^{\frac{-(t-c)}{\sqrt{2}}}\cos\left(\frac{t-c}{\sqrt{2}}\right)\right)\,dt\\
=&\frac{8\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-a}{\sqrt{2}}\right)+\cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}-\\
&\frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}-\\
&\frac{8\mathrm{e}^{-\frac{a}{\sqrt{2}}}\left(\sin\left(\frac{a}{\sqrt{2}}\right)-\cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+\\
&\frac{16\sqrt{2}a-\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(10\sin\left(\frac{c}{\sqrt{2}}\right)-\left(2\sqrt{2}c-10\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)-8}{16\sqrt{2}}
\end{aligned}
\tag{B.12.28}
$$

□

Derivation B.12.5. *Let K be a Silverman's kernel with CDF as in Def B.12.1 and partial L2-product of the CDF as in B.12.4. The partial L2-product of the CCDF at two different points centre points 0 and $c \in \mathbb{R}$ is*

1. For $c \geq 0$

$$
\xi_K(a,c) = \begin{cases}
\begin{aligned}
&\frac{8\mathrm{e}^{-\frac{c-a}{\sqrt{2}}}\left(-\sin\left(\frac{c-a}{\sqrt{2}}\right)+\cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+\\
&\frac{\mathrm{e}^{-\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)-\cos\left(\frac{c-2a}{\sqrt{2}}\right)-2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+\\
&\frac{8\mathrm{e}^{\frac{a}{\sqrt{2}}}\left(\sin\left(\frac{a}{\sqrt{2}}\right)+\cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}+\\
&\frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(10\sin\left(\frac{c}{\sqrt{2}}\right)-\left(2\sqrt{2}+10\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)-8-16\sqrt{2}a}{16\sqrt{2}}
\end{aligned} & \textit{for} \quad a \in (-\infty, 0] \\
\begin{aligned}
&\frac{\mathrm{e}^{\frac{-c}{\sqrt{2}}}\left(-2\sin\left(\frac{c-2a}{\sqrt{2}}\right)+5\sin\left(\frac{c}{\sqrt{2}}\right)-\left(2\sqrt{2}(c-a)+5\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}}-\\
&\frac{8\mathrm{e}^{\frac{-a}{\sqrt{2}}}\left(\sin\left(\frac{a}{\sqrt{2}}\right)-\cos\left(\frac{a}{\sqrt{2}}\right)\right)}{16\sqrt{2}}
\end{aligned} & \textit{for} \quad a \in [0, c] \\
\frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for} \quad a\,[c, \infty)
\end{cases}
$$

2. *For* $c \leq 0$

$$\xi_K(a,c) = \begin{cases} \frac{8\mathrm{e}^{-\frac{c-a}{\sqrt{2}}}\left(-\sin\left(\frac{c-a}{\sqrt{2}}\right)+\cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + & \\ \frac{\mathrm{e}^{-\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)-\cos\left(\frac{c-2a}{\sqrt{2}}\right)-2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} + & \\ \frac{8\mathrm{e}^{\frac{a}{\sqrt{2}}}\Big((\sin\left(\frac{a}{\sqrt{2}}\right)-\cos\left(\frac{a}{\sqrt{2}}\right)\Big)}{16\sqrt{2}} + & \\ \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(-10\sin\left(\frac{c}{\sqrt{2}}\right)+(2\sqrt{2}c-10)\cos\left(\frac{c}{\sqrt{2}}\right)\right)-16\sqrt{2}(a-c)}{16\sqrt{2}} & \textit{for} \quad a \in (-\infty, c] \\ \frac{8\mathrm{e}^{\frac{c-a}{\sqrt{2}}}\left(\sin\left(\frac{c-a}{\sqrt{2}}\right)+\cos\left(\frac{c-a}{\sqrt{2}}\right)\right)}{16\sqrt{2}} - & \\ \frac{\mathrm{e}^{\frac{c}{\sqrt{2}}}\left(2\sin\left(\frac{c-2a}{\sqrt{2}}\right)+5\sin\left(\frac{c}{\sqrt{2}}\right)-\left(2\sqrt{2}a-5\right)\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for} \quad a \in [c, 0] \\ \frac{\mathrm{e}^{\frac{c-2a}{\sqrt{2}}}\left(\sin\left(\frac{c-2a}{\sqrt{2}}\right)+\cos\left(\frac{c-2a}{\sqrt{2}}\right)+2\cos\left(\frac{c}{\sqrt{2}}\right)\right)}{16\sqrt{2}} & \textit{for} \quad a \in [0, \infty] \end{cases}$$

Proof. From Lemma 5.4.2, $\xi_K(a,c)$ is the reflection of $\gamma_K(a,c)$ on the y - axis. Hence, we can compute $\xi_K(a,c)$ by taking $\gamma_K(a,c)$. □

B.13 Computation Mean Absolute Deviation for Gaussian

Here, we compute the mean absolute (MAE) deviation for Gaussian

1. Standard Gaussian (mean 0 and standard deviation a)
2. Gaussian distribution with mean μ and standard deviation σ
3. Gaussian kernel distribution (applicable for Gaussian mixture)

Derivation B.13.1. *Let Y be a random variable from a standard Gaussian distribution, i.e. $Y \sim \mathcal{N}(0,1)$. Then, the MAE for Y is*

$$\mathbb{E}|Y| = \sqrt{\frac{2}{\pi}}$$

Proof.

$$\mathbb{E}|Y| = -\int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}} y\mathrm{e}^{-\frac{y^2}{2}}\,dy + \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}} y\mathrm{e}^{-\frac{y^2}{2}}\,dy = \frac{2}{\sqrt{2\pi}}$$

□

Derivation B.13.2. *Let Y be a random variable from a Gaussian distribution with mean μ and standard deviation σ, i.e. $Y \sim \mathcal{N}(\mu, \sigma)$. Then, the MAE for Y is*

$$\mathbb{E}|Y| = \mu\,\mathrm{erf}\left(\frac{\mu}{\sigma\sqrt{2}}\right) + \frac{\sqrt{2}\sigma}{\sqrt{\pi}}\mathrm{e}^{-\frac{-\mu^2}{2\sigma^2}}$$

Proof.

$$\begin{aligned}\mathbb{E}|Y| = &-\int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}} y\mathrm{e}^{-\frac{(y-\mu)^2}{2\sigma^2}}\,dy + \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}} y\mathrm{e}^{-\frac{(y-\mu)^2}{2\sigma^2}}\,dy \\ =&\mu\,\mathrm{erf}\left(\frac{\mu}{\sigma\sqrt{2}}\right) + \frac{\sqrt{2}\sigma}{\sqrt{\pi}}\mathrm{e}^{-\frac{-\mu^2}{2\sigma^2}}\end{aligned}$$

□

Derivation B.13.3. *Suppose a random sample Y be a random variable with mean $\mu_i, \ldots, \mu_N$ and standard deviation $\sigma_1, \ldots, \sigma_N$ with weight function $w_1, \ldots, w_N$.*

The MAE is

$$\mathbb{E}|Y| = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(\frac{\mu_{ij}\pi \operatorname{erf}\left(\frac{\mu_{ij}}{\sigma_{ij}\sqrt{2}}\right)}{\pi} + \frac{2\sigma_{ij} \mathrm{e}^{-\frac{\mu_{ij}^2}{2\sigma_{ij}^2}}}{\sqrt{2\pi}} \right)$$

Proof.

$$\begin{aligned}\mathbb{E}|Y| =& \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(\int_{-\infty}^{0} -\frac{1}{\sqrt{2\pi}\sqrt{\sigma_i^2+\sigma_j^2}} y \mathrm{e}^{-\frac{y-(\mu_i-\mu_j)^2}{2\sqrt{\sigma_i^2+\sigma_j^2}}}\, dy + \right.\\ &\left. \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\sqrt{\sigma_i^2+\sigma_j^2}} y \mathrm{e}^{-\frac{y-(\mu_i-\mu_j)^2}{2\sqrt{\sigma_i^2+\sigma_j^2}}}\, dy \right)\end{aligned}$$

Let $\mu_{ij} = \mu_i + \mu_j$, $\sigma_{ij}^2 = \sigma_i^2 + \sigma_j^2$ and $\sigma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2}$

$$\mathbb{E}|Y| = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(\int_{-\infty}^{0} -\frac{1}{\sqrt{2\pi}\sqrt{\sigma_{ij}^2}} y \mathrm{e}^{-\frac{y-(\mu_{ij})^2}{2\sqrt{\sigma_{ij}^2}}}\, dy + \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{ij}^2}} y \mathrm{e}^{-\frac{y-(\mu_{ij})^2}{2\sqrt{\sigma_{ij}^2}}}\, dy \right).$$

Using integral by substitution,

$$\begin{aligned} z = \frac{y-\mu_{ij}}{\sigma_{ij}} &\qquad y = z\sigma_{ij} + \mu_{ij} \\ \frac{dz}{dy} = \frac{1}{\sigma_{ij}} &\qquad dy = \sigma_{ij}\, dz \end{aligned}$$

$$\begin{aligned}\mathbb{E}|Y| =& \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(-\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{\mu_{ij}}{\sigma_{ij}}} (z\sigma_{ij}+\mu_{ij})\mathrm{e}^{-\frac{z^2}{2}}\, dz + \frac{1}{\sqrt{2\pi}} \int_{-\frac{\mu_{ij}}{\sigma_{ij}}}^{\infty} (z\sigma_{ij}+\mu_{ij})\mathrm{e}^{-\frac{z^2}{2}}\, dz \right)\\ =& \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(-\left(\frac{\pi\mu - \pi\mu_{ij}\operatorname{erf}\left(\frac{\mu_{ij}}{\sqrt{2}\sigma_{ij}}\right)}{2\pi} - \frac{\sigma_{ij}\mathrm{e}^{-\frac{\mu_{ij}^2}{2v^2}}}{\sqrt{2\pi}} \right) + \right.\\ &\left. \left(\frac{\sigma_{ij}\mathrm{e}^{-\frac{\mu_{ij}^2}{2v^2}}}{\sqrt{2\pi}} + \frac{\pi\mu_{ij}\operatorname{erf}\left(\frac{\mu_{ij}}{\sqrt{2}\sigma_{ij}}\right) + \pi\mu_{ij}}{2\pi} \right) \right)\\ =& \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \left(\frac{\mu_{ij}\pi \operatorname{erf}\left(\frac{\mu_{ij}}{\sigma_{ij}\sqrt{2}}\right)}{\pi} + \frac{2\sigma_{ij} \mathrm{e}^{-\frac{\mu_{ij}^2}{2\sigma_{ij}^2}}}{\sqrt{2\pi}} \right)\end{aligned}$$

□

Appendix C

Chapter 6: Investigation of Tuning for Distribution Estimation

To support the main proof of our investigation, we list down preliminaries lemmas that will be useful.

C.1 List of Definitions

Definition C.1.1. *Let $f : D \to \mathbb{R}$ be a function defined on an open interval D such that $D \subseteq \mathbb{R}$. Let $a \in \bar{D}$. We say that*

$$\lim_{x \to a} f(x) = \infty$$

if for every integer $c_1 > 0$ there is some number $x_1 > 0$ such that

$$f(x) > c_1$$

whenever $0 < |x - a| < x_1$.

Definition C.1.2. *Let $f : \bar{\mathbb{R}} \to \mathbb{R}$ be a function defined on $\bar{\mathbb{R}} = \{\mathbb{R} \cup -\infty, \infty\}$. We say that*

$$\lim_{x \to \infty} f(x) = \infty$$

if for every integer $c_2 > 0$ there is some number $x_2 > 0$ such that

$$f(x) > c_2$$

whenever $x > x_2$.

Definition C.1.3. *Let $f : D \to \mathbb{R}$ be a function defined on an open interval D such that $D \subseteq \mathbb{R}$. Let $a \in \bar{D}$. We say that*

$$\lim_{x \to a} f(x) = -\infty$$

if for every negative integer $c_3 < 0$ there is some number $x_3 > 0$ such that

$$f(x) < c_3$$

whenever $0 < |x - a| < x_3$.

C.2 List of Lemmas

Lemma C.2.1. *Consider the function $g_1 : \mathbb{R}^+ \to \mathbb{R}$,*

$$g_1(x) = \frac{a}{x} \exp\left\{\frac{-a}{x^2}\right\}. \tag{C.2.1}$$

where $a \in \mathbb{R}^+$. Then, $\lim\limits_{x \to \infty} g_1 = 0$.

Proof. For Eqn (C.2.1), by using the properties of limit,

$$\lim_{x \to \infty} \frac{a}{x} \exp\left\{\frac{-a}{x^2}\right\} = \lim_{x \to \infty} \frac{a}{x} \lim_{x \to \infty} \exp\left\{\frac{-a}{x^2}\right\} \tag{C.2.2}$$

$$= 0 \times 0 = 0. \tag{C.2.3}$$

Hence, $\lim\limits_{x \to \infty} g_1 = 0$.

□

Lemma C.2.2. *Consider the function $g_2 : \mathbb{R}^+ \to \mathbb{R}$,*

$$g_2(x) = -\frac{a}{x^2} \exp\left\{-\frac{a}{x^2}\right\}. \tag{C.2.4}$$

where $a \in \mathbb{R}^+$. Then, $\lim\limits_{x\to 0} g_2(x) = -\infty$.

Proof. Re-write the above as

$$g_2(x) = \frac{-\frac{a}{x}}{\frac{1}{\exp\left\{\frac{-a}{x^2}\right\}}}. \tag{C.2.5}$$

By L'Hopital rule, differentiate the numerator and denominator w.r.t x,

$$-\frac{x}{2} \exp\left\{-\frac{a}{x^2}\right\}. \tag{C.2.6}$$

Then,

$$\lim_{x\to 0} -\frac{x}{2} \exp\left\{-\frac{a}{x^2}\right\} = -\infty. \tag{C.2.7}$$

□

Lemma C.2.3. *Let $g_2 : \mathbb{R}^+ \to \mathbb{R}$ be a function of x, be a function of x, such that*

$$g_3(x) = \exp\left\{-\frac{a}{x^2}\right\}. \tag{C.2.8}$$

where $a \in \mathbb{R}^+$. Then, $\lim\limits_{x\to 0} g_3(x) = 0$.

Proof. Re-write the above as

$$g_3(x) = \frac{1}{\exp\left\{\frac{a}{x^2}\right\}}. \tag{C.2.9}$$

As $x \to 0$, $\frac{a}{x^2} \to \infty$, i.e $\lim\limits_{x\to 0} \frac{a}{x^2} = \infty$. By the elementary property of exponential function $\mathrm{e}^{\infty} = \infty$. Hence, $\lim\limits_{x\to 0} \frac{1}{\mathrm{e}^{\frac{a}{x^2}}} = 0$ which is equal to $\lim\limits_{x\to 0} \mathrm{e}^{-\frac{a}{x^2}} = 0$. □

Bibliography

[1] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[4] Frithjof Gressmann, Franz J. Király, Bilal Mateen, and Harald Oberhauser. Probabilistic supervised learning, 2019.

[5] B.W Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[6] Charles J Stone et al. An asymptotically optimal window selection rule for kernel density estimates. *The Annals of Statistics*, 12(4):1285–1297, 1984.

[7] Raphael Sonabend and Franz Kiraly. distr6: R6 object-oriented probability distributions interface in r. *arXiv preprint arXiv:2009.02993*, 2020.

[8] Sheldon M Ross. *A first course in probability / Sheldon Ross*. Pearson Education, Upper Saddle River, N.J., 8th ed. edition, 2010. ISBN 9780136079095.

[9] M.C Jones. The performance of kernel density functions in kernel distribution function estimation. 9(2):129–132, 1990. ISSN 0167-7152.

[10] Yi-Hung Kung, Pei-Sheng Lin, and Cheng-Hsiung Kao. An optimal k-nearest neighbor for density estimation. *Statistics & Probability Letters*, 82 (10):1786 – 1791, 2012. ISSN 0167-7152.

[11] IJ Good. The probabilistic explication of information, evidence, surprise, causality, explanation, and utility. *Foundations of statistical inference*, pages 108–141, 1971.

[12] IJ Good. Non-parametric roughness penalty for probability densities. *Nature physical science*, 229(1):29–30, 1971.

[13] I. J. Good and R. A. Gaskins. Nonparametric roughness penalties for probability densities. *Biometrika*, 58(2):255–277, 1971. ISSN 00063444.

[14] B. W. Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *Ann. Statist.*, 10(3):795–810, 09 1982. doi: 10.1214/aos/1176345872. URL `https://doi.org/10.1214/aos/1176345872`.

[15] Tom Leonard. Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40 (2):113–132, 1978.

[16] Chong Gu and Chunfu Qiu. Smoothing spline density estimation: Theory. *The Annals of Statistics*, pages 217–234, 1993.

[17] Chong Gu and Jingyuan Wang. Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica*, pages 811–826, 2003.

[18] Charles Kooperberg and Charles J. Stone. Logspline density estimation for censored data. *Journal of Computational and Graphical Statistics*, 1(4):301–328, 1992. ISSN 10618600. URL `http://www.jstor.org/stable/1390786`.

[19] Peter Hall, Prakash Patil, et al. On wavelet methods for estimating smooth functions. *Bernoulli*, 1(1-2):41–58, 1995.

[20] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[21] Shouhong Wang. A neural network method of density estimation for univariate unimodal data. *Neural Computing & Applications*, 2(3):160–167, 1994.

[22] Malik Magdon-Ismail and Amir F Atiya. Neural networks for density estimation. In *Advances in Neural Information Processing Systems*, pages 522–528, 1999.

[23] Parikshit Ram and Alexander G Gray. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–635. ACM, 2011.

[24] Kun Yang and Wing Hung Wong. Density estimation via adaptive partition and discrepancy control. *arXiv preprint arXiv:1404.1425*, 2014.

[25] Linxi Liu and Wing Hung Wong. *Multivariate density estimation based on adaptive partitioning: Convergence rate, variable selection and spatial adaptation*. Department of Statistics, Stanford University, 2014.

[26] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[27] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[28] Tan Nguyen and Scott Sanner. Algorithms for direct 0–1 loss optimization in binary classification. In *International Conference on Machine Learning*, pages 1085–1093. PMLR, 2013.

[29] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[30] Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, 3, 2005.

[31] Allan H Murphy. Scalar and vector partitions of the probability score: Part i. two-state situation. *Journal of Applied Meteorology*, 11(2):273–282, 1972.

[32] Andreas P Weigel, Mark A Liniger, and Christof Appenzeller. The discrete brier and ranked probability skill scores. *Monthly Weather Review*, 135(1): 118–124, 2007.

[33] Edward S Epstein. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology (1962-1982)*, 8(6):985–987, 1969.

[34] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pages 231–238, 2000.

[35] Thomas G Dietterich and Eun Bae Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. 1995.

[36] Leo Breiman. Bias, variance, and arcing classifiers. 1996.

[37] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

[38] Robert Tibshirani. *Bias, variance and prediction error for classification rules*. Citeseer, 1996.

[39] Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[40] AP Dawid. Encyclopedia of statistical sciences 2., 1984.

[41] Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *Metron*, 72(2):169–183, 2014.

[42] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[43] Christophe Giraud-Carrier. Metalearning-a tutorial. In *Tutorial at the 7th international conference on machine learning and applications (ICMLA), San Diego, California, USA*, 2008.

[44] Joaquin Vanschoren. Understanding machine learning performance with experiment databases. *lirias. kuleuven. be, no*, 2010.

[45] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44(1):117–130, 2015.

[46] Pavel Brazdil and Christophe Giraud-Carrier. Metalearning and algorithm selection: progress, state of the art and introduction to the 2018 special issue, 2018.

[47] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[48] Saharon Rosset and Eran Segal. Boosting density estimation. In *Advances in Neural Information Processing Systems*, pages 657–664, 2003.

[49] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[50] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[51] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[52] Greg Ridgeway. Generalized boosted models: A guide to the gbm package, 2005.

[53] Peter Bühlmann and Bin Yu. Boosting with the l 2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

[54] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[55] David W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, Inc, second edition edition, 2015.

[56] IJ Good. Probability and the weighing of evidence,(london: C. griffin, 1950).,". *Rational Decisions," JRSS, Ser. B*, 14:107–114, 1952.

[57] J. D. F Habbema, J Hermans, and K van den. Broek Leiden. A stepwise discriminant analysis program using density estimation. *In Compstat*, 1974.

[58] Robert P. W. Duin. On the choice of smoothing parameters for parzen estimators of probability density functions. *IEEE Transactions on Computers*, (11):1175–1179, 1976.

[59] James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976. ISSN 00251909, 15265501. URL `http://www.jstor.org/stable/2629907`.

[60] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102 (477):359–378, 2007.

[61] Daniel Friedman. Effective scoring rules for probabilistic forecasts. *Management Science*, 29(4):447–454, 1983.

[62] Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9(2):65–78, 1982. ISSN 03036898, 14679469. URL `http://www.jstor.org/stable/4615859`.

[63] Adrian Bowman, Peter Hall, and Tania Prvan. Bandwidth selection for the smoothing of distribution functions. *Biometrika*, 85(4):799–808, 1998.

[64] Peter Hall. On kullback-leibler loss and density estimation. *The Annals of Statistics*, 15(4):1491–1519, 1987. ISSN 00905364.

[65] Adrian W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984. ISSN 00063444. URL `http://www.jstor.org/stable/2336252`.

[66] Naomi Altman and Christian Léger. Bandwidth selection for kernel distribution function estimation. *Journal of Statistical Planning and Inference*, 46(2):195 – 214, 1995. ISSN 0378-3758. doi: https://doi.org/10.1016/0378-3758(94)00102-2. URL `http://www.sciencedirect.com/science/article/pii/0378375894001022`.

[67] Byeong U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990. ISSN 01621459. URL `http://www.jstor.org/stable/2289526`.

[68] Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):683–690, 1991.

[69] PETER Hall, SIMON J. Sheather, M. C. Jones, and J. S. Marron. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika*, 78 (2):263–269, 1991. doi: 10.1093/biomet/78.2.263. URL `+http://dx.doi.org/10.1093/biomet/78.2.263`.

[70] Bruce E Hansen. Bandwidth selection for nonparametric distribution estimation. *manuscript, University of Wisconsin*, 2004.

[71] Padhraic Smyth and David H. Wolpert. Stacked density estimation. In *NIPS*, 1997.

[72] Xubo Song, Kun Yang, and Misha Pavel. Density boosting for gaussian mixtures. In Nikhil Ranjan Pal, Nik Kasabov, Rajani K. Mudi, Srimanta Pal, and Swapan Kumar Parui, editors, *Neural Information Processing*, pages 508–515, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30499-9.

[73] Jingyi Cui, Hanyuan Hang, Yisen Wang, and Zhouchen Lin. Gbht: Gradient boosting histogram transform for density estimation. *arXiv preprint arXiv:2106.05738*, 2021.

[74] Robert A Vandermeulen, René Saitenmacher, and Alexander Ritchie. A proposal for supervised density estimation.

[75] Kaiyuan Wu, Wei Hou, and Hongbo Yang. Density estimation via the random forest method. *Communications in Statistics-Theory and Methods*, 47(4): 877–889, 2018.

[76] Aristidis Likas. Probability density estimation using artificial neural networks. *Computer physics communications*, 135(2):167–175, 2001.

[77] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[78] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[79] Mathias Bourel and Badih Ghattas. Aggregating density estimators: an empirical study. *arXiv preprint arXiv:1207.4959*, 2012.

[80] Marco Di Marzio and Charles C. Taylor. Boosting kernel density estimates: A bias reduction technique? *Biometrika*, 91(1):226–233, 2004. ISSN 00063444. URL `http://www.jstor.org/stable/20441091`.

[81] Marco Di Marzio and Charles C Taylor. On boosting kernel density methods for multivariate data: density estimation and classification. *Statistical Methods and Applications*, 14(2):163–178, 2005.

[82] Mathias Bourel and Jairo Cugliari. Bagging of density estimators. *arXiv preprint arXiv:1808.03447*, 2018.

[83] Padhraic Smyth and David Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1):59–83, 1999.

[84] M. C. JONES, O. LINTON, and J. P. NIELSEN. A simple bias reduction method for density estimation. *Biometrika*, 82(2):327–338, 06 1995. ISSN 0006-3444. doi: 10.1093/biomet/82.2.327. URL `https://doi.org/10.1093/biomet/82.2.327`.

[85] Aleksandra Baszczyńska et al. Kernel estimation of cumulative distribution function of a random variable with bounded support. *Statistics in Transition. New Series*, 17(3):541–556, 2016.

[86] Michel Lejeune and Pascal Sarda. Smooth estimators of distribution and density functions. *Computational Statistics & Data Analysis*, 14(4):457–471, 1992.

[87] Adriano Z Zambom and Dias Ronaldo. A review of kernel density estimation with applications to econometrics. *International Econometric Review*, 5(1):20–42, 2013.

[88] Peter Hall. Cross-validation in density estimation. *Biometrika*, 69(2):383–390, 1982.

[89] George R. Terrell. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85(410):470–477, 1990. ISSN 01621459. URL `http://www.jstor.org/stable/2289786`.

[90] Pascal Sarda. Smoothing parameter selection for smooth distribution functions. *Journal of Statistical Planning and Inference*, 35(1):65–75, 1993.

[91] M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American statistical association*, 91(433):401–407, 1996.

[92] Matt P Wand and M Chris Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2):97–116, 1994.

[93] David W. Scott and George R. Terrell. Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association*, 82 (400):1131–1146, December 1987. ISSN 0162-1459.

[94] David W Scott, Richard A Tapia, James R Thompson, et al. Nonparametric probability density estimation by discrete maximum penalized-likelihood criteria. *Annals of Statistics*, 8(4):820–832, 1980.

[95] George R Terrell and David W Scott. Oversmoothed nonparametric density estimates. *Journal of the American Statistical Association*, 80(389):209–214, 1985.

[96] MC Jones and SJ Sheather. Using non-stochastic terms to advantage in kernel-based estimation of integrated squared density derivatives. *Statistics & Probability Letters*, 11(6):511–514, 1991.

[97] Alan M Polansky and Edsel R Baker. Multistage plug—in bandwidth selection for kernel distribution function estimates. *Journal of Statistical Computation and Simulation*, 65(1-4):63–80, 2000.

[98] Peter Hall and James Stephen Marron. Extent to which least-squares cross-validation minimises integrated square error in nonparametric density estimation. *Probability Theory and Related Fields*, 74(4):567–581, 1987.

[99] J Steve Marron and Matt P Wand. Exact mean integrated squared error. *The Annals of Statistics*, pages 712–736, 1992.

[100] Francisco J Goerlich Gisbert. Weighted samples, kernel density estimators and convergence. *Empirical Economics*, 28(2):335–351, 2003.

[101] Bin Wang and Xiaofeng Wang. Bandwidth selection for weighted kernel density estimation. *arXiv preprint arXiv:0709.1616*, 2007.

[102] Dariush Ghorbanzadeh, Philippe Durand, and Luan Jaupi. A method for the generate a random sample from a finite mixture distributions. In *International Conference on Computational Mathematics, Computational Geometry & Statistics (CMCGS). Proceedings*, page 1. Global Science and Technology Forum, 2017.

[103] Alexander Jordan, Fabian Krüger, and Sebastian Lerch. Evaluating probabilistic forecasts with scoringrules. *arXiv preprint arXiv:1709.04743*, 2017.

[104] Hans Hersbach. Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems. *Weather and Forecasting*, 15(5): 559–570, 10 2000. ISSN 0882-8156. doi: 10.1175/1520-0434(2000)

015⟨0559:DOTCRP⟩2.0.CO;2. URL https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2.

[105] Guillem Candille and Olivier Talagrand. Evaluation of probabilistic prediction systems for a scalar variable. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(609):2131–2150, 2005.

[106] Steven V Weijs, Ronald Van Nooijen, and Nick Van De Giesen. Kullback–leibler divergence as a forecast skill score with classic reliability–resolution–uncertainty decomposition. *Monthly Weather Review*, 138(9):3387–3399, 2010.

[107] Ludwig Baringhaus and Carsten Franz. On a new multivariate two-sample test. *Journal of multivariate analysis*, 88(1):190–206, 2004.

[108] Gabor J Szekely, Maria L Rizzo, et al. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of classification*, 22(2):151–184, 2005.

[109] Eric P Grimit, Tilmann Gneiting, Veronica J Berrocal, and Nicholas A Johnson. The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 132(621C):2925–2942, 2006.

[110] William F Trench. *Introduction to real analysis*. Open textbook library. San Antonio, Tex, 2013. ISBN 0130457868. URL https://open.umn.edu/opentextbooks/BookDetail.aspx?bookId=174.

[111] Uci machine learning repository. URL a.

[112] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL https://www.R-project.org/.

[113] Raphael Sonabend, Franz J Király, Andreas Bender, Bernd Bischl, and Michel Lang. mlr3proba: An R Package for Machine Learning in Survival Analysis. *Bioinformatics*, 02 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab039. URL https://doi.org/10.1093/bioinformatics/btab039. btab039.

[114] Raphael Sonabend and Franz Kiraly. distr6: R6 object-oriented probability distributions interface in r. *arXiv preprint arXiv:2009.02993*, 2020.

[115] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[116] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[117] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[118] Ken Arnold, James Gosling, and David Holmes. *The Java programming language*. Addison Wesley Professional, 2005.

[119] Michel Lang, Martin Binder, Jakob Richter, Patrick Schratz, Florian Pfisterer, Stefan Coors, Quay Au, Giuseppe Casalicchio, Lars Kotthoff, and Bernd Bischl. mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software*, dec 2019. doi: 10.21105/joss.01903. URL `https://joss.theoj.org/papers/10.21105/joss.01903`.

[120] Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones. mlr: Machine learning in r. *Journal of Machine Learning Research*, 17(170):1–5, 2016. URL `https://jmlr.org/papers/v17/15-066.html`.

[121] Max Kuhn and Hadley Wickham. *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.*, 2020. URL `https://www.tidymodels.org`.

[122] Max Kuhn. The caret package. *R Foundation for Statistical Computing, Vienna, Austria. URL https://cran. r-project. org/package= caret*, 2012.

[123] Omar Trejo Navarro. *R Programming By Example: Practical, Hands-on Projects to Help You Get Started with R*. Packt Publishing, 2017. ISBN 1788292545.

[124] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL `https://CRAN.R-project.org/doc/Rnews/`.

[125] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011. URL `https://www.jstatsoft.org/v39/i05/`.

[126] Max Kuhn. A short introduction to the caret package. *R Found Stat Comput*, 1, 2015.

[127] Hadley Wickham. *Advanced r*. CRC press, 2019.

[128] David Lucy, Robert Aykroyd, and Maintainer David Lucy. Package 'genkern'. 2012.

[129] Alejandro Quintela del Rio, Graciela Estevez Perez, and Maintainer Alejandro Quintela del Rio. Package 'kerdiest'. 2012.

[130] Tarn Duong. ks: Kernel smoothing. r package version 1.9. 3. *URL: http://CRAN. R-project. org/package= ks*, 2014.

[131] AW Bowman and A Azzalini. R package 'sm': Nonparametric smoothing methods (version 2.2–4) http://www. stats. gla. ac. uk/, adrian/sm. *Last Accessed: Oct*, 30:2012, 2010.

[132] Adrian W Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, volume 18. OUP Oxford, 1997.

[133] Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the r package tda. *arXiv preprint arXiv:1411.1830*, 2014.

[134] Joachim Engel, Eva Herrmann, and Theo Gasser. An iterative bandwidth selector for kernel estimation of densities and their derivatives. *Journaltitle of Nonparametric Statistics*, 4(1):21–34, 1994.

[135] Tristen Hayfield and Jeffrey S Racine. Nonparametric econometrics: The np package. *Journal of statistical software*, 27(5):1–32, 2008.

[136] Qi Li and Jeff Racine. Nonparametric estimation of distributions with categorical and continuous data. *journal of multivariate analysis*, 86(2):266–292, 2003.

[137] Chong Gu. Smoothing spline ANOVA models: R package gss. *Journal of Statistical Software*, 58(5):1–25, 2014. URL http://www.jstatsoft.org/v58/i05/.

[138] Christian Schellhase. R-package pendensity-density estimation with a penalized mixture approach. 2019.

[139] Charles Kooperberg. logspline: Logspline density estimation routines. *R package version*, 2(9), 2016.

[140] Florian Camphausen, Matthias Kohl, Peter Ruckdeschel, Thomas Stabla, R Core Team, and Maintainer Peter Ruckdeschel. Package 'distr'. *Cran. org. Version*, 2, 2019.

[141] Alex Hayes and Ralph Moller. distributions3: Probability distributions as s3 objects. 2019. URL https://cran.r-project.org/package=distributions3.

[142] Lukas Sablica and Kurt Hornik. mistr: A Computational Framework for Mixture and Composite Distributions. *The R Journal*, 12(1):283–299, 2020. doi: 10.32614/RJ-2020-003. URL https://journal.r-project.org/archive/2020/RJ-2020-003/index.html.

[143] Shaiful Anuar Abu Bakar, Saralees Nadarajah, Zahrul Azmir ABSL Kamarul Adzhar, and Ibrahim Mohamed. Gendist: An R Package for Generated Probability Distribution Models. *PLOS ONE*, 11(6):1–20, June 2016. doi: 10.1371/journal.pone.0156. URL https://ideas.repec.org/a/plo/pone00/0156537.html.

[144] Torsten Hothorn, Friedrich Leisch, Achim Zeileis, and Kurt Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699, 2005.

[145] Manuel JA Eugster. Benchmark experiments a tool for analyzing statistical learning algorithms. 2011.

[146] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[147] Bin Liu, Ying Yang, Geoffrey I Webb, and Janice Boughton. A comparative study of bandwidth choice in kernel density estimation for naive bayesian classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 302–313. Springer, 2009.

[148] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[149] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[150] Simon J. Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004. ISSN 08834237. URL `http://www.jstor.org/stable/4144429`.

[151] Omar M Eidous, Mohammad Abd Alrahem Shafeq Marie, and Mohammed H Ebrahem. A comparative study for bandwidth selection in kernel density estimation. *Journal of Modern Applied Statistical Methods*, 9(1):26, 2010.

[152] Ibrahim A Ahmad and Iris S Ran. Data based bandwidth selection in kernel density estimation with parametric start via kernel contrasts. *Journal of Nonparametric Statistics*, 16(6):841–877, 2004.

[153] Greg Ridgeway. Looking for lumps: boosting and bagging for density estimation. *Computational Statistics & Data Analysis*, 38(4): 379 – 392, 2002. ISSN 0167-9473. doi: https://doi.org/10.1016/S0167-9473(01)00066-4. URL `http://www.sciencedirect.com/science/article/pii/S0167947301000664`. Nonlinear Methods and Data Mining.

[154] Nishtha Hooda, Seema Bawa, and Prashant Singh Rana. Fraudulent firm classification: a case study of an external audit. *Applied Artificial Intelligence*, 32(1):48–64, 2018.

[155] D. D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, and Y. Zhang. Failure analysis of parameter-induced simulation crashes in climate models. *Geoscientific Model Development*, 6(4):1157–1171, 2013. doi: 10.5194/gmd-6-1157-2013. URL `https://gmd.copernicus.org/articles/6/1157/2013/`.

[156] I-Cheng Yeh. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and concrete composites*, 29(6): 474–480, 2007.

[157] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[158] Michael Scott Brown, Michael J Pelosi, and Henry Dirska. Dynamic-radius species-conserving genetic algorithm for the financial forecasting of dow jones index stocks. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 27–41. Springer, 2013.

[159] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

[160] Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[161] Davide Chicco and Giuseppe Jurman. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC medical informatics and decision making*, 20(1):16, 2020.

[162] Oguz Akbilgic, Hamparsum Bozdogan, and M Erdal Balaban. A novel hybrid rbf neural networks model as a forecaster. *Statistics and Computing*, 24 (3):365–375, 2014.

[163] Pedro FB Silva, Andre RS Marcal, and Rubim M Almeida da Silva. Evaluation of features for leaf discrimination. In *International Conference Image Analysis and Recognition*, pages 197–204. Springer, 2013.

[164] Clara Higuera, Katheleen J Gardiner, and Krzysztof J Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6):e0129126, 2015.

[165] Max A Little, Patrick E McSharry, Stephen J Roberts, Declan AE Costello, and Irene M Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical engineering online*, 6(1):23, 2007.

[166] Matteo Cassotti, Davide Ballabio, Viviana Consonni, Andrea Mauri, Igor V Tetko, and Roberto Todeschini. Prediction of acute aquatic toxicity toward daphnia magna by using the ga-k nn method. *Alternatives to Laboratory Animals*, 42(1):31–41, 2014.

[167] Francesca Grisoni, Viviana Consonni, Sara Villa, Marco Vighi, and Roberto Todeschini. Qsar models for bioconcentration: Is the increase in the complexity justified by more accurate predictions? *Chemosphere*, 127:171–179, 2015.

[168] Francesca Grisoni, Viviana Consonni, Marco Vighi, Sara Villa, and Roberto Todeschini. Investigating the mechanisms of bioconcentration through qsar classification trees. *Environment international*, 88:198–205, 2016.

[169] M Cassotti, D Ballabio, R Todeschini, and V Consonni. A similarity-based qsar model for predicting acute toxicity towards the fathead minnow (pimephales promelas). *SAR and QSAR in Environmental Research*, 26(3): 217–243, 2015.

[170] I-Cheng Yeh and Tzu-Kuang Hsu. Building real estate valuation models with comparative approach through case-based reasoning. *Applied Soft Computing*, 65:260–271, 2018.

[171] Małgorzata Charytanowicz, Jerzy Niewczas, Piotr Kulczycki, Piotr A Kowalski, Szymon Łukasik, and Sławomir Żak. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*, pages 15–24. Springer, 2010.

[172] Alejandro Quintela del Rio, Graciela Estevez Perez, and Maintainer Alejandro Quintela del Rio. Package ‘kerdiest’. 2012.

[173] Tarn Duong et al. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21(7): 1–16, 2007.

[174] Adrian Bowman, Adelchi Azzalini, Maintainer Adrian Bowman, and TRUE LazyData. Package 'sm', 2018.

[175] Tristen Hayfield and Jeffrey S Racine. Nonparametric econometrics: The np package. *Journal of statistical software*, 27(5):1–32, 2008.

[176] Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the r package tda. *arXiv preprint arXiv:1411.1830*, 2014.

[177] Christian Schellhase and Göran Kauermann. Density estimation and comparison with a penalized mixture approach. *Computational Statistics*, 27(4): 757–777, 2012.

[178] Christian Schellhase. R-package pendensity-density estimation with a penalized mixture approach. 2019.

[179] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[180] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[181] Peter Bjorn Nemenyi. *Distribution-free multiple comparisons*. Princeton University, 1963.

[182] Edgar Acuna and Alex Rojas. Bagging classifiers based on kernel density estimators. In *Proceedings of the International Conference on New Trends in Computational Statistics with Biomedical Applications*, pages 343–350, 2001.