Research on Positive-Unlabeled Learning

XIAOKE WANG

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

May 21, 2024

I, XIAOKE WANG, 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 work.

Abstract

Positive-unlabeled (PU) learning handles classification tasks on the data containing only labeled-positive instances and unlabeled instances. PU learning has been applied in many fields of observational studies. Support vector machine (SVM)-based PU learning is one of the main branches of PU learning and offers a range of advantages, e.g., the efficiency of training and the generalisation ability. Moreover, the SVM-based PU classifiers are able to generate non-linear decision boundary by employing kernel trick to capture complex relationships among features and have been shown to achieve robust performance. This study focuses on SVM-based PU classifiers and contains three contributions. Firstly we proposed global and local PU classifier with asymmetric loss (GLPUAL) with kernel trick applied for satisfactory classification on trifurcated PU datasets, where the positive set is constituted by two subsets distributing on both sides of the negative set. Secondly, to address the unsatisfactory interpretability and performance of GLPUAL on the PU datasets containing irrelevant features, we introduced L_1-norm regularisation to the objective function of GLPUAL to construct a sparse classifier to remove irrelevant features. The proposed classifier is termed elastic GLPUAL (E-GLPUAL). Then a kernel-free technique was introduced to E-GLPUAL to generate non-linear decision boundary. The proposed classifier is termed elastic kernel-free GLPUAL (EKF-GLPUAL). Thirdly, we proposed class-prior-based GLPUAL (CPB-GLPUAL) by introducing a technique of unbiased PU learning to GLPUAL for better performance when the class prior is known. Besides, we explored the conditions for CPB-GLPUAL to exhibit universal consistency between the 0-1 classification risk of CPB-GLPUAL and the Bayes risk.

Impact Statement

Positive-unlabeled (PU) learning is a branch of semi-supervised learning, where only a certain amount of positive instances are labeled in the dataset. PU learning has received increasing attention in recent years. Our work focuses on binary PU classification and is expected to be applicable to fields such as deceptive review detection and text categorization in future.

In Chapter 3, firstly we proposed a new classifier, which is termed global and local PU classifier with asymmetric loss (GLPUAL), for better performance on the PU datasets where the distances from the two positive subsets to the ideal decision boundary are very different. Secondly, we introduced the kernel trick to GLPUAL to generate non-linear decision boundary in the original feature space for satisfactory performance on trifurcated datasets, where there are two subsets of the positive set distributing on both sides of the negative set.

In Chapter 4, firstly we noticed that irrelevant features are hard to be removed by GLPUAL. Motivated by this, we proposed elastic GLPUAL (E-GLPUAL) by introducing a L1-norm regularised term into the objective function of GLPUAL to assign zero coefficients to the irrelevant features on PU datasets. Secondly, for E-GLPUAL to generate non-linear decision boundary, we introduced the kernel free techniques from the soft quadratic surface SVM (SQSSVM) to the objective function of E-GLPUAL.

In Chapter 5, firstly we proposed Class-Prior-Based GLPUAL (CPB-GLPUAL), where there is one fewer hyper-parameter to be tuned than GLPUAL for better generalization ability, and classification. The two hyper-parameters critical to classification can be determined by the class prior. Secondly, we introduced the kernel

trick to CPB-GLPUAL to generate non-linear decision boundary in the original feature space for better classification.

Acknowledgements

First and foremost, I am deeply grateful to my supervisor, Prof. Jinghao Xue. Frankly speaking, I am neither intelligent nor resilient as a student. Without Professor Xue's selfless guidance and encouragement throughout my PhD journey, urging me not to give up on my ideals, I would certainly not have been able to complete this long journey.

Secondly, I would also like to express my gratitude to my second supervisor, Dr. Rui Zhu, for her invaluable advice and guidance during the last four years. Furthermore, I am extremely grateful to Dr. Xiaochen Yang for her patient advise for the revision of my thesis.

Thirdly, I would like to express my gratitude to Weihao Xia, Hai Wang, and Huiling Zheng for their various forms of assistance during my academic pursuits. From them, I have learned many admirable qualities and invaluable spirits.

Lastly, I would like to apologize to the people mentioned above for the troubles I have caused them in both academic and personal matters. I am grateful again for everyone's continued tolerance and acceptance to me.

Contents

1 Introduction **14**

1.1 Background 14

1.2 Notations in this thesis 15

1.3 Important Abbreviations in this Thesis 16

2 Literature Review **17**

2.1 Multi-Step Methods 17

2.2 One-Step Methods 18

2.2.1 Inconsistent PU Learning Methods 18

2.2.2 Consistent PU Learning Methods 19

2.3 Details of the Important Methods for this Research 20

2.3.1 Biased Support Vector Machine (BSVM) 20

2.3.2 Biased Least-Squares Support Vector Machine (BLS-SVM) 21

2.3.3 Global and Local Learning Classifier (GLLC) 22

2.3.4 Unbiased PU Learning (uPU) 23

2.3.5 Non-Negative PU Learning (nnPU) for Balanced Data . . . 24

3 Global and Local PU Classifier with Asymmetric Loss (GLPUAL) **25**

3.1 Introduction 25

3.2 Methodology of GLPUAL for Linear Decision Boundary in the Original Feature Space 28

3.3 Algorithm of GLPUAL for Linear Decision Boundary in the Original Feature Space 29

3.3.1 Alternating Direction Method of Multipliers (ADMM) . . . 29

3.3.2 Update of β and β_0 31

3.3.3 Update of h 32

3.4 Experiments on Synthetic Data 33

3.4.1 The Generation of Synthetic Positive-Negative (PN) Datasets 33

3.4.2 Training-Test Split for the Synthetic PU Datasets 34

3.4.3 Model Setting 34

3.4.4 Results and Analysis 35

3.5 Kernel Trick to GLPUAL for Non-Linear Decision in the Original Feature Space 36

3.6 Experiments on Real Data 40

3.6.1 The Source of Real Datasets 40

3.6.2 Training-Test Split for the Real PU Datasets 41

3.6.3 Compared Methods 42

3.6.4 Model Setting 42

3.6.5 Summary of Results 43

3.6.6 Pattern Analysis for the Real Datasets Preferring GLPUAL to GLLC 44

3.6.7 Pattern Analysis for the Real Datasets Preferring GLLC to GLPUAL 45

3.6.8 Results among GLPUAL, uPU and nnPU 47

3.7 Conclusion 48

4 Elastic GLPUAL (E-GLPUAL) and Elastic Kernel Free GLPUAL (EKF-GLPUAL) 49

4.1 Introduction 49

4.2 Methodology and Algorithm of E-GLPUAL for Linear Decision Boundary 50

4.2.1 Update of β and β_0 53

4.2.2 Update of h 54

4.2.3 Update of a 55

4.2.4 Update of t . . . 55

4.2.5 Update of b . . . 56

4.3 Experiments for E-GLPUAL on Synthetic Datasets . . . 56

4.3.1 The Generation of Synthetic PN Datasets . . . 57

4.3.2 Training-Test Split for the Synthetic PU Datasets . . . 57

4.3.3 Model Setting . . . 57

4.3.4 Results and Analysis . . . 58

4.4 Methodology and Algorithm of EKF-GLPUAL for Quadratic Decision Boundary . . . 59

4.4.1 Update of α^* and β_0 . . . 63

4.4.2 Update of h . . . 64

4.4.3 Update of a . . . 65

4.4.4 Update of b . . . 65

4.4.5 Update of t . . . 66

4.5 Experiments for EKF-GLPUAL on Real Datasets . . . 67

4.5.1 The Source of Datasets . . . 67

4.5.2 The Preprocessing of the Datasets . . . 67

4.5.3 Training-Test Split for the Real PU Datasets . . . 68

4.5.4 Compared Methods and Model Setting . . . 68

4.5.5 Results and Analysis . . . 69

4.6 Some Theoretical Exploration of on Grouping Effect . . . 71

4.7 Conclusion . . . 76

5 Class-Prior-Based GLPUAL (CPB-GLPUAL) 77

5.1 Introduction . . . 77

5.2 Assumption for CPB-GLPUAL . . . 78

5.2.1 Case-Control Scenario . . . 78

5.2.2 Known Class Prior . . . 78

5.3 Methodology of CPB-GLPUAL for Linear Decision Boundary in the Original Feature Space . . . 79

5.4 Algorithm of CPB-GLPUAL for Linear Decision Boundary in the Original Feature Space . . . 81
5.4.1 ADMM for Non-Convex Optimisation . . . 81
5.4.2 Update of a . . . 84
5.4.3 Update of h . . . 87
5.4.4 Update of β and β_0 . . . 88
5.5 Experiments for CPB-GLPUAL on Synthetic Datasets . . . 89
5.5.1 The Generation of Synthetic PN Datasets . . . 89
5.5.2 Training-Test Split for the Synthetic PU Datasets . . . 89
5.5.3 Model Setting . . . 89
5.5.4 Results and Analysis . . . 90
5.6 Kernel Trick to CPB-GLPUAL for Non-Linear Decision Boundary . 90
5.7 Experiments for CPB-GLPUAL on Real Datasets . . . 94
5.7.1 The Source of Datasets and Experimental Design . . . 94
5.7.2 Training-Test Split for the Real PU Datasets . . . 94
5.7.3 Compared Methods and Model Setting . . . 95
5.7.4 Results and Analysis . . . 95
5.8 Universal Consistency of CPB-GLPUAL . . . 97
5.8.1 Introduction to Bayes Risk . . . 97
5.8.2 Universal Consistency of CPB-GLPUAL . . . 98
5.8.3 An Approximate of CPB-GLPUAL to Simplify the Proof of Universal Consistency . . . 99
5.8.4 Three Steps of the Proof of Theorem 3 . . . 103
5.8.5 Step 1: Construction of the 'Representative' Dataset . . . 103
5.8.6 Step 2: Proof of Theorem 4 by Contradiction . . . 107
5.8.7 Step 3: Proof of Theorem 3 . . . 118
5.8.8 Discussion on Universal Consistency of CPB-GLPUAL. . . 119
5.9 Conclusions . . . 119

6 Conclusions and Future Work 121

6.1 Conclusions . . . 121

6.2 Limitations and Implications 122
6.3 Future Work 124

Bibliography **125**

List of Figures

3.1 The 2-dimensional projection with t-SNE of dataset **wifi** 26

3.2 A potential pattern of the constructed linearly separable space converted from the original trifurcated PU datasets via the kernel trick. . 27

3.3 The difference between the hinge loss and the squared loss. 27

3.4 The decision boundaries trained by (left) GLPUAL and (right) GLLC on the synthetic data. 37

3.5 Boxplots for the difference between F1-scores of GLPUAL and GLLC on each dataset with label frequency $\gamma = 0.5$. 44

3.6 Boxplots for the difference between F1-scores of GLPUAL and GLLC on each dataset with label frequency of 0.25. 44

3.7 The t-SNE plots of the best four datasets 45

3.8 The t-SNE plots of the worst five datasets 46

4.1 Illustration of the steps to construct the jth row of matrix $\mathbb{X}_i$. 61

5.1 The similarity between the absolute loss and the squared loss. 82

5.2 The four cases of the threshold function $j_p|x| + \frac{1}{2}(x - d_p)^2, j_p < 0$. . 86

6.1 Flowchart to determine which method mentioned in this thesis to use. 123

List of Tables

1.1 Summary of notations. 15

1.2 Important abbreviations in this thesis 16

3.1 The average percentage F1-score on the synthetic dataset. 35

3.2 Summary of the datasets for the verification of the motivation of GLPUAL. 41

3.3 The average F1-score (%) of the classifiers trained on the 16 real PU datasets 47

4.1 The average F1-score on the synthetic dataset. 58

4.2 The proportion of the irrelevant features whose parameters were compressed to zero. 59

4.3 Summary of the datasets for the verification of the motivation of EKF-GLPUAL. 68

4.4 The average F1-score (%) of the classifiers trained on the 14 processed real PU datasets and their standard deviations 70

4.5 The proportion of the irrelevant features whose parameters were compressed to zero. 71

5.1 The average F1-score on the synthetic dataset. 90

5.2 Summary of the datasets used in experiments for the evaluation of CPB-GLPUAL. 94

5.3 The average F1-score of the classifiers trained on the 16 real PU datasets and their standard deviations 96

Chapter 1

Introduction

1.1 Background

Current works on positive-unlabeled (PU) learning mainly focus on binary scenarios [1], and thus the classifiers stated in this thesis are regarded to be binary by default. PU learning is to train a classifier to distinguish between positive and negative instances on PU data, which only contain labeled-positive instances and unlabeled instances; this indicates that PU data lack the information for negative labels while generally a training set for machine learning should have label information for both positive labels and negative labels.

For example in practice, if we aim to train a classifier based on YouTube backend data to discern whether a user likes a particular YouTuber (positive: like; negative: dislike) , subscribers can be regarded to be the users who like this YouTuber and hence can be treated as the positive instances. However, can we confidently assert that users who did not subscribe dislike the YouTuber? Clearly not. Some users may merely lack the habit of subscribing or may have subscribed the YouTuber on alternative platforms but not on YouTube. Therefore, hastily categorizing these non-subscribed users as negatives without taking any action can introduce significant bias into the trained classifier, subsequently impacting video recommendation strategies.

Moreover, there are more and more PU data occurring in other areas, such as deceptive review detection [2], text categorization [3] and remote sensing classification [4, 5]. The raising demand for accurate classification on PU data yields the development of PU learning, and hence this thesis.

1.2 Notations in this thesis

The notations used in this thesis are summarised in Table 1.1.

Table 1.1: Summary of notations.

Symbol	Description
p, u, n	Indicators of labeled-positive set, unlabeled set and negative set.
n_p, n_u, n_n, n_{pu}	Size of the labeled-positive set, unlabeled set, negative set, and the whole dataset.
m	The number of attributes contained in an instance.
$\boldsymbol{x}$	Column vector of attributes of an instance.
$\boldsymbol{X}_{[p]}, \boldsymbol{X}_{[u]}, \boldsymbol{X}_{[pu]}$	Data matrix of attributes of labeled-positive set, unlabeled set and the whole training set.
y	The class of the instances; $y = 1$: positive; $y = -1$: negative.
$\boldsymbol{y}_{[p]}, \boldsymbol{y}_{[u]}, \boldsymbol{y}_{[pu]}$	The class of the instances in positive set, unlabeled set and the whole training set.
$\boldsymbol{Y}_{[p]}, \boldsymbol{Y}_{[u]}, \boldsymbol{Y}_{[pu]}$	Diagonal matrix with diagonal elements $\boldsymbol{y}_{[p]}$, $\boldsymbol{y}_{[u]}$ and $\boldsymbol{y}_{[pu]}$, respectively.
$(\boldsymbol{X}, Y)$	An unknown instance of attributes $\boldsymbol{X}$ and class Y regarded as random variables.
$\boldsymbol{P}, \boldsymbol{N}$	The conditional distribution of $\boldsymbol{X}$ in the positive set and negative set.
π	Class prior $P[Y = 1]$.
γ	Label frequency, i.e., the proportion of positive instances to be labeled in the dataset.
f	The predictive score function of a binary classifier for PU data.
$(\boldsymbol{\beta}, \beta_0)$	The vector of the parameters of the classifier to be trained.
J	The objective function of the classifier to be trained.
l	Loss function of the classifier to be trained.
L	Expected loss, i.e., the expectation of loss function l.
$\mathscr{L}, \mathscr{L}_a$	Lagrangian function and augmented Lagrangian function of the objective function.
$\lambda, c, c_p, c_u, C_p, C_u$	Hyper-parameters in GLLC and GLPUAL while $C_p = \frac{1}{n_p} c_p, C_u = \frac{1}{n_u} c_u$.
$\boldsymbol{R}$	A similarity matrix of the instances.
s	Labeling indicator of an instance; $s = 1$: labeled (-positive); $s = -1$: unlabeled.
$\mathbf{1}$	A column with all elements as 1.
$\boldsymbol{h}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{t}$	Block variables for ADMM.
$\boldsymbol{u}_h, \boldsymbol{u}_a, \boldsymbol{v}, \boldsymbol{q}$	Lagrangian variables.
$\boldsymbol{\phi}\{\boldsymbol{a}, \boldsymbol{b}\}$	Kernel transform for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$; if $\boldsymbol{a} = \boldsymbol{b}$, denoted as $\boldsymbol{\phi}\{\boldsymbol{a}\}$.
$\boldsymbol{\Phi}\{\boldsymbol{A}, \boldsymbol{B}\}$	A matrix of the kernel transform for the rows of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$; if $\boldsymbol{A} = \boldsymbol{B}$, denoted as $\boldsymbol{\Phi}\{\boldsymbol{A}\}$.

1.3 Important Abbreviations in this Thesis

The important abbreviations and their full names used in this thesis are summarised in Table 1.2

Table 1.2: Important abbreviations in this thesis

Full Name	Abbreviation
positive-unlabeled	PU
positive-negative	PN
unbiased PU Learning	uPU
non-negative PU Learning	nnPU
elastic GLPUAL	E-GLPUAL
elastic kernel free GLPUAL	EKF-GLPUAL
class-prior-based GLPUAL	CPB-GLPUAL
global and local PU classifier with asymmetric loss	GLPUAL
soft quadratic surface SVM	SQSSVM
t-distributed stochastic neighbor embedding	t-SNE
alternating direction method of multipliers	ADMM
cross-validation	CV
radial basis function	RBF
Karush-Kuhn-Tucker conditions	KKT conditions
principal component analysis	PCA

Chapter 2

Literature Review

This chapter introduces previous works on PU learning, including the multi-step methods and one-step methods. The multi-step methods generally apply several classifiers for PU learning for its final output, while one-step methods only apply one classifier [1]. The work done in this study is mainly based on the one-step biased PU learning methods.

2.1 Multi-Step Methods

The main difficulty in PU learning is the lack of labeled-negative instances. A natural way to deal with this issue is to pick the instances highly likely to be negative from the unlabeled set and regard them as negative. In this case, the obtained dataset will then contain positive, negative and unlabeled instances. Then PU learning can be converted to semi-supervised learning and thus semi-supervised classifiers can be applied to this dataset.

The multi-step methods following this idea are often referred as two-step methods. The first step of two-step methods is to select reliable negative instances from the unlabeled set, contributing to a dataset for semi-supervised learning; techniques for this step include Spy [6], 1-DNF [7], Rocchio [8] and PGPU [9]. The second step of two-step methods is to train a semi-supervised classifier to label the rest of the unlabeled instances, e.g. by using DILCA-KNN [10] and TFIPNDF [11]. Subsequently, the two step methods are generalised to have more steps by regarding the output of the second step as the pseudo labels for further iteratively training

[12, 13, 14, 15, 16].

The main advantage of the multi-step methods is that it constructs an embedded framework allowing us to solve PU learning problem via the achievements on semi-supervised learning. However, the accuracy of multi-step methods relies heavily on the accuracy of the algorithm applied in the first step to pick reliable negative instances [17].

2.2 One-Step Methods

One-step methods can be further categorized into inconsistent PU learning methods and consistent PU learning methods, depending on if the objective function is a consistent estimator of the expectation of a certain loss to classify an unknown instance form the population.

2.2.1 Inconsistent PU Learning Methods

Motivated by the issue that treating all the unlabeled instances as negative can impose bias to the training of the classifiers, the inconsistent PU learning methods were proposed to alleviate the bias in these ’naive’ classifiers. The objective function of these methods cannot be considered as a consistent estimator of the expectation of a certain loss on an unknown instance for classification; this is why we collectively refer to these methods as inconsistent PU learning methods.

An early attempt of the inconsistent PU learning methods was the biased support vector machine (BSVM) [18] based on the classic supervised support vector machine (SVM) [19], assigning high weight to the average loss of the labeled-positive instances and low weight to the average loss of the unlabeled instances in the objective function. Subsequently, weighted unlabeled samples SVM (WUS-SVM) was proposed in [20] to assign a unique weight to each of the unlabeled instances according to the likelihood of this unlabeled instance to be negative. Then, the biased least squares SVM (BLSSVM) was proposed in [21] by substituting the squared loss for the hinge loss in the objective of BSVM in case that too much importance is given to the unlabeled-positive instances during the training of the classifier. Meanwhile, also in [21], the local constraint was introduced to the objective function of BLSSVM, encouraging the instances to be classified into the same class as its neighborhoods and the proposed method is the global and local learning classifier (GLLC). Moreover, the large-margin label-calibrated SVM (LLSVM) was proposed in [22] to further alleviate the bias by introducing hat loss to the objective function, where the hat loss measures the gap between the arc-tan of the predictive score function (without intercept) and a certain threshold.

2.2.2 Consistent PU Learning Methods

The consistent PU learning methods were proposed to minimise the risk of the classification of an unknown instance, thus the objective function of the consistent PU learning methods was designed to be a consistent estimator of the expectation of a certain loss on an unknown instance for classification.

A pioneer consistent PU learning method is the unbiased PU learning (uPU) proposed in [23], whose objective function is an unbiased and consistent estimator of the expectation of a certain loss on an unknown instance for classification. Subsequently, the non-negative PU learning (nnPU) was proposed in [24] by taking the absolute value of the estimated average loss on the negative set in the objective function of uPU for convergence of the classifier training. Furthermore, for better performance on the imbalanced PU training set, imbalanced nnPU (imbalancednnPU) was proposed in [25] to make the objective function equivalent to a consistent estimator of the expectation of a certain loss on an unknown instance from the balanced population for classification.

Then a new framework of the objective function of the PU classifiers was proposed in [26] based on the variational principle, where the class prior is not needed for the classifier training while both uPU and nnPU need the class prior as the prior knowledge. For the similar aim, a Taylor series expansion-based variational framework named T-HOneCls was proposed in [27] based on the Taylor variational loss. Then a gradient-based regulariser was introduced into the objective function of nnPU in [5] to ensure the proposed classifier, named GradPU, can function effectively in the case of labeling bias. In [28], the pinball loss factorization and centroid smoothing (Pin-LFCS) was proposed by introducing the pinball loss to the objective function for robust classification when there is noise in the features.

According to the above mentioned methods in this section, the probability of every positive instance in the PU dataset to be labeled is fixed and the same under the selected completely at random (SCAR) assumption. However, there are also PU datasets where the positive instances are selected to be labeled with various probability, leading to the selected at random (SAR) assumption. The performance of the classifiers constructed under the SCAR assumption is likely to be unsatisfactory on the PU dataset consistent to the SAR assumption. In this case PU learning with a Selection Bias (PUSB) was proposed in [29] by applying a score function to the objective function of nnPU to handle the selection bias problems in uPU and nnPU. Then based on the SAR assumption, [30] proposed the labeling bias estimation (LBE) via the graphic method to enhance the performance of the non-linear deep model with a multi-layer perceptron on PU dataset with labeling bias.

2.3 Details of the Important Methods for this Research

In this section, more details of the important methods for this research are revealed.

2.3.1 Biased Support Vector Machine (BSVM)

Recall the objective function of the classic SVM as

$$\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \sum_{\boldsymbol{x}_i^T \in \boldsymbol{X}_{pn}} [1 - y_i(\boldsymbol{x}_i\boldsymbol{\beta} + \beta_0)]_+, \tag{2.1}$$

where $\boldsymbol{X}_{pn}$ denotes the dataset with all instances correctly labeled and λ is a positive hyper-parameter. The objective function in Equation 2.1 is to find the hyperplane with the maximum margin to separate positive and negative instances. In this case, SVM can only applied for supervised learning.

In order to enable SVM to handle PU learning classification, BSVM treats all the unlabeled instances as negative and assign the loss of labeled-positive instances and the loss of unlabeled instances with different weights in the objective function. BSVM trains classifiers by solving the following objective function:

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0}\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}+C_p\mathbf{1}_p^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]_+ \\ +C_u\mathbf{1}_u^T[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]_+,\end{aligned} \tag{2.2}$$

where $\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_m)^T\in\mathbb{R}^{m\times 1}$ is the vector of the model parameters, $C_p=\frac{1}{n_p}c_p$, $C_u=\frac{1}{n_u}c_u$ are positive hyper-parameters, $[g(\cdot)]_+$ indicates the column vector of the maximum between each element of $g(\cdot)$ and 0, and $\mathbf{1}_{p,u}=(\underbrace{1,1,\cdots,1}_{k})^T, k=n_p,n_u$. The predictive score function obtained by BSVM is the same as the predictive score function of SVM, i.e.,

$$f=\boldsymbol{x}_i^T\boldsymbol{\beta}+\beta_0. \tag{2.3}$$

2.3.2 Biased Least-Squares Support Vector Machine (BLS-SVM)

One weakness of BSVM is that sometimes the hinge loss in the objective function of BSVM in Equation 2.2 selects more unlabeled-positive instances than the unlabeled-negative instances to be the support vectors for the negative class, which constructs a decision boundary tending to misclassify the unlabeled-positive instances as negative. This is more likely to happen when there are many unlabeled-positive instances close to the unlabeled negative set.

To deal with this issue, the BLS-SVM for the PU learning was proposed by [31] to force all training instances to contribute to the construction of the decision boundary of the trained SVM by solving the following optimisation:

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0}\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}+C_p[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)] \\ +C_u[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]^T[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)],\end{aligned} \tag{2.4}$$

where the squared loss replaces the hinge loss applied in BSVM on both labeled-positive set and unlabeled set. The objective function of BLS-SVM makes all the instances contribute to the construction of the decision boundary hence the importance given to the unlabeled-positive instances is restricted [32].

The predictive score function obtained by BLS-SVM is also the same as the predictive score function of SVM in Equation 2.3.

2.3.3 Global and Local Learning Classifier (GLLC)

The similarities between a training instance and its neighbors can also be treated as a factor for classification, which is named local learning [33]. It is noted in [21] that the gap between PU learning and classical supervised learning on accuracy can be mitigated via GLLC.

GLLC is a combination of BLS-SVM and local learning. The objective function of GLLC is given as

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0}\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} &+ C_p[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]\\ &+C_u[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]^T[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]\\ &+(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0),\end{aligned} \tag{2.5}$$

where $\mathbf{1}_{pu}=(\underbrace{1,1,\cdots,1}_{k})^T, k=n_p+n_u$. $\boldsymbol{R}$ is the similarity matrix for the instances and their neighbors, which can be obtained in Equation 2.7.

To obtain the similarity matrix $\boldsymbol{R}$, we firstly need to calculate matrix $\boldsymbol{W}$ by

$$w_{ij}=\begin{cases}\exp\left(-\sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)^T(\boldsymbol{x}_i-\boldsymbol{x}_j)\right) & \text{if the } i\text{th and } j\text{th instances are KNN of each other,}\\ 0 & \text{otherwise,}\end{cases} \tag{2.6}$$

where σ is a hyper-parameter to be selected.

Let $\boldsymbol{w}_{\cdot i}$ denote the ith column of matrix $\boldsymbol{W}$ and $\boldsymbol{W}^*$ denote a diagonal matrix with $d_{ii}=\mathbf{1}_{[pu]}^T\boldsymbol{w}_{\cdot i}$, one can obtain

$$\boldsymbol{R}=\frac{1}{(n_p+n_u)}(\boldsymbol{W}^*-\boldsymbol{W}). \tag{2.7}$$

The predictive score function obtained by GLLC is also the same as the predictive score function of SVM in Equation 2.3.

2.3.4 Unbiased PU Learning (uPU)

The objective function of uPU was designed to be an unbiased estimator of the expected loss to classify an unknown instance under the SCAR assumption. The main advantage of uPU is that the weight hyper-parameters in its objective function are determined by class prior $\pi = P[Y=1]$.

To obtain the objective function of uPU, one can firstly let random variables $(\boldsymbol{X}, Y)$ denote an unknown instance of attributes $\boldsymbol{X}$ and Class Y and suppose that there is a predictive score function $f(\boldsymbol{X};\boldsymbol{\beta})$ of a PU classifier. The loss function for f is defined as $l(f,y)$, where $y = -1, 1$. Let $\boldsymbol{P}$ denote the distribution of $\boldsymbol{X}$ in the positive set and $\boldsymbol{N}$ denote the distribution of $\boldsymbol{X}$ in the negative set. According to the law of total expectation, the expected loss $L(f)$ of the predictive score function on a new instance is given as

$$L(f) = \mathbb{E}[l(f(\boldsymbol{X};\boldsymbol{\beta}),Y)] = \pi L_p^1(f) + (1-\pi)L_n^{-1}(f), \tag{2.8}$$

where $L_p^1(f) = \mathbb{E}_{\boldsymbol{X}\sim\mathbf{P}}[l(f(\boldsymbol{X};\boldsymbol{\beta}),1)]$ and $L_n^{-1}(f) = \mathbb{E}_{\boldsymbol{X}\sim\mathbf{N}}[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)]$.

Then consider

$$P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)] = P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)|Y=1]\pi + P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)|Y=-1](1-\pi). \tag{2.9}$$

Hence we have

$$P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)|Y=-1](1-\pi) = P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)] - P[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)|Y=1]\pi. \tag{2.10}$$

Taking expectation w.r.t. $\boldsymbol{X}$ using the probabilities at both sides of Equation

2.10, one can obtain

$$(1-\pi)L_n^{-1}(f) = L_u^{-1}(f) - \pi L_p^{-1}(f), \tag{2.11}$$

where $L_p^{-1}(f) = \mathbb{E}_{X\sim\mathbf{P}}[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)]$ and $L_u^{-1}(f) = \mathbb{E}[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)]$.

Combining Equation 2.8 and Equation 2.11, there is

$$L(f) = \pi L_p^1(f) + L_u^{-1}(f) - \pi L_p^{-1}(f). \tag{2.12}$$

Therefore the objective function for uPU is

$$\min_{\boldsymbol{\beta}} \pi \hat{L}_p^1(f) + \hat{L}_u^{-1}(f) - \pi \hat{L}_p^{-1}(f), \tag{2.13}$$

where $\hat{L}_p^1(f) = \frac{1}{n_p}\sum_{\boldsymbol{x}\in p} l(f(\boldsymbol{X};\boldsymbol{\beta}),1)$, $\hat{L}_u^{-1}(f) = \frac{1}{n_u}\sum_{\boldsymbol{x}\in u} l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)$ and $\hat{L}_p^{-1}(f) = \frac{1}{n_p}\sum_{\boldsymbol{x}\in p} l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)$. In this case, $\hat{L}_p^1(f)$, $\hat{L}_u^{-1}(f)$ and $\hat{L}_p^{-1}(f)$ are unbiased estimators of $L_p^1(f)$, $L_u^{-1}(f)$ and $L_p^{-1}(f)$, respectively. The weights of the loss in the objective function of uPU in Equation 2.13 are determined by class prior π.

2.3.5 Non-Negative PU Learning (nnPU) for Balanced Data

In the objective function of uPU in Equation 2.13, negative value of $\hat{L}_u^{-1}(f) - \pi\hat{L}_p^{-1}(f)$ might appear and this can cause the algorithm of uPU, which is based on ADAM [34], unable to converge to the optimal solution.[24].

As a remedy of this issue, nnPU was hence proposed in [24] by simply substituting $\hat{L}_u^{-1}(f) - \pi\hat{L}_p^{-1}(f)$ with $\max\{\hat{L}_u^{-1}(f) - \pi\hat{L}_p^{-1}(f), 0\}$ in the objective function of uPU in Equation 2.13 as

$$\min_{\boldsymbol{\beta}} \pi \hat{L}_p^1(f) + \max\{\hat{L}_u^{-1}(f) - \pi \hat{L}_p^{-1}(f), 0\}. \tag{2.14}$$

Chapter 3

Global and Local PU Classifier with Asymmetric Loss (GLPUAL)

3.1 Introduction

Recall the objective function of GLLC:

$$
\begin{aligned}
\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} &+ C_p[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)] \\
&+ C_u[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\
&+ (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0),
\end{aligned}
\tag{3.1}
$$

where the squared loss for the classifier is applied to both the labeled-positive set and the unlabeled set. The similarity matrix $\boldsymbol{R}$ is obtained via Equation 2.6 and Equation 2.7.

There is a special kind of PU datasets where the positive set is constituted by two subsets distributing on the both sides of the negative set. We call this kind of PU datasets as trifurcated PU datasets. To visualise the pattern of the trifurcated PU datasets with more than two attributes, we can utilise the t-distributed stochastic neighbor embedding (t-SNE) [35], which was proposed for non-linear dimensional reduction. An example of the trifurcated PU datasets is illustrated in the following 2-dimensional projection with t-SNE of dataset **wifi** [36] in Figure 3.1, which will be further discussed in Section 3.6. The classifiers with the non-linear decision boundary

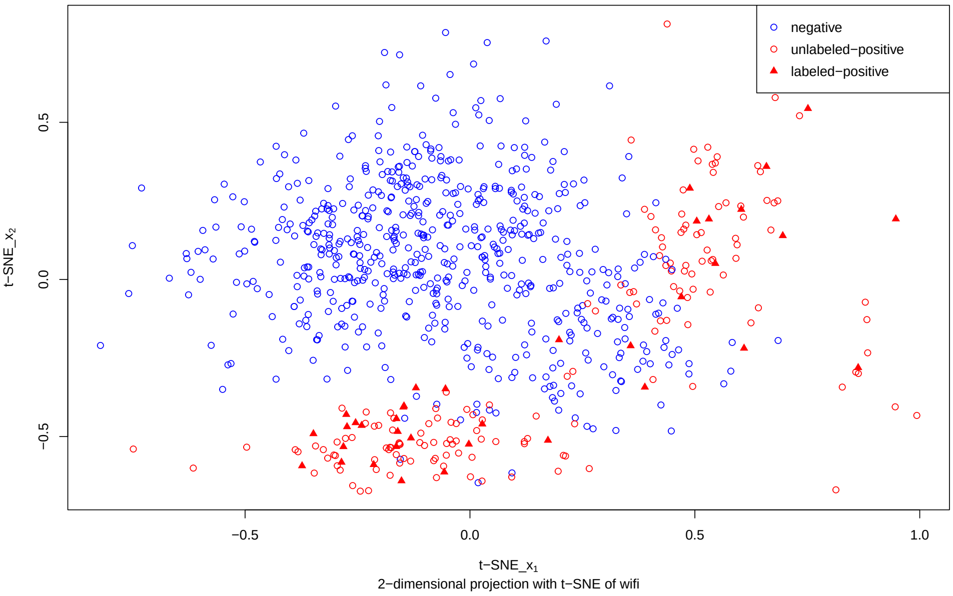

Figure 3.1: The 2-dimensional projection with t-SNE of dataset **wifi**, where the positive set is constituted by two subsets distributing on the both sides of the negative set.

in the original feature space are needed for the classification on the trifurcated PU datasets. However, when we applied kernel trick on GLLC to obtain the non-linear decision boundary in the original feature space, the satisfactory performance still cannot be achieved.

It should be noted that the essence of GLLC with kernel trick is to train a classifier with linear decision boundary in a linearly separable space constructed by a certain mapping of the original space. In this case, one of the potential reasons for the unsatisfactory performance of GLLC on the trifurcated PU datasets is that in the constructed linearly separable space, the original trifurcated PU datasets is converted to follow the pattern in Figure 3.2, where the distances from the two positive subsets to the ideal decision boundary, as indicated by a solid blue line, are very different. Moreover, as shown in Figure 3.3, the squared loss in GLLC can impose quadratic penalty not only on the instances wrongly classified but also on all the instances correctly classified. Therefore, the labeled-positive instances correctly classified by the ideal boundary but far away form the ideal decision boundary, as circled by the dashed lines in Figure 3.2, can generate large penalty via the squared loss and

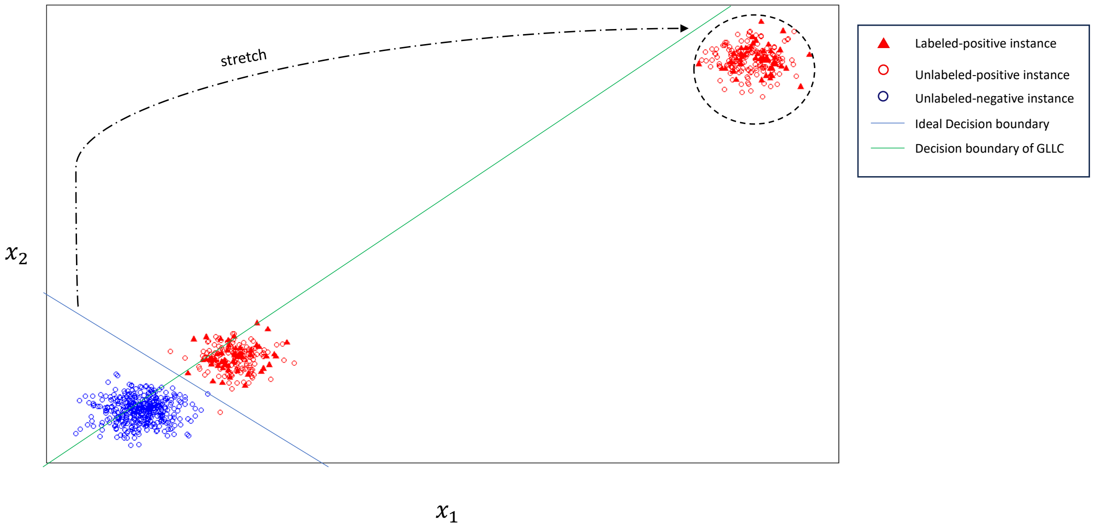

Figure 3.2: A potential pattern of the constructed linearly separable space converted from the original trifurcated PU datasets via the kernel trick; x_1 and x_2 represent the mappings of the features in the original trifurcated PU dataset.

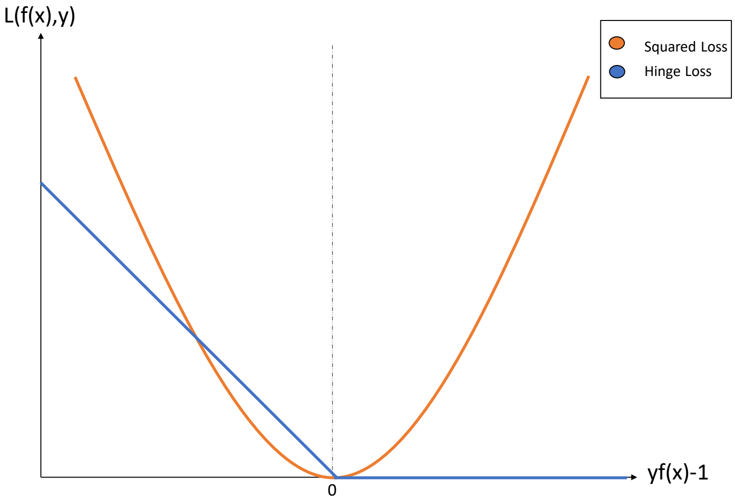

Figure 3.3: The difference between the hinge loss and the squared loss; x-axis: the distance between the instance and its correct margin; the negative distance indicates that the instance lies on the wrong side of margin while the positive distance indicates that the instance lies on the correct side of the margin; y-axis: the loss of the predictive score function f.

hence stretches the ideal decision boundary towards the labeled-positive instances far away. This leads to the optimal decision boundary of GLLC in green solid line, which misclassified more instances than the ideal decision boundary.

The aim of applying the squared loss in the objective function of GLLC is to ensure the importance given to the unlabeled-positive instances to be negative support vectors to be lower than the importance given to unlabeled-negative instances to be negative support vectors, hence it is not necessary to also use the squared loss on the labeled-positive set. Furthermore, as shown in the right side to zero of x-axis Figure 3.3, the hinge loss does not penalise the instances lying on the correct side of the margin. Hence with the hinge loss applied, the instances correctly classified but far away from the ideal decision boundary will not generate any penalty.

Motivated by the above analysis, we proposed Global and Local PU Classifier with Asymmetric Loss (GLPUAL) for better classification on the trifurcated PU datasets. Firstly, in Section 3.2 and Section 3.3, we proposed the methodology and algorithm of GLPUAL to generate the linear decision boundary much more close to the ideal decision boundary for the datasets following the pattern in Figure 3.2 but in the original feature space. This was achieved by using the hinge loss for the labeled-positive instances and the squared loss for the unlabeled-instances. Secondly in Section 3.4, we conducted experiments on simple synthetic datasets to access the performance of GLPUAL. Thirdly, we introduced the kernel trick to GLPUAL to generate the non-linear decision boundary in the original feature space in Section 3.5. In this case, GLPUAL can be applied on the trifurcated PU datasets and the experiments on 16 real datasets, including 4 trifurcated PU datasets, were conducted to verify our motivation in Section 3.6.

3.2 Methodology of GLPUAL for Linear Decision Boundary in the Original Feature Space

Suppose that there are n_p labeled-positive instances and n_u unlabeled instances with m features. Let feature matrix $\boldsymbol{X}_{[pu]} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_p}, \ldots, \boldsymbol{x}_{n_p+n_u})^T \in \mathbb{R}^{(n_p+n_u)\times m}$, where the column vector $\boldsymbol{x}_i \in \mathbb{R}^{m\times 1}$ denotes the vector of the features of the ith instance. Similarly, matrix $\boldsymbol{X}_{[p]} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_p})^T \in \mathbb{R}^{n_p\times m}$ denotes the feature matrix of the labeled-positive set while matrix $\boldsymbol{X}_{[u]} = (\boldsymbol{x}_{n_p+1}, \boldsymbol{x}_{n_p+2}, \ldots, \boldsymbol{x}_{n_p+n_u})^T \in \mathbb{R}^{n_u\times m}$ denotes the feature matrix of the unlabeled set. According to [21], we use the labeling indicator s_i as the pseudo label of the instance $\boldsymbol{x}_i$. Then the unconstrained optimisation problem of GLPUAL is formulated as

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0} & \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C_p\mathbf{1}_p^T\left[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)\right]_+ \\ & + C_u[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\ & + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0),\end{aligned} \tag{3.2}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T \in \mathbb{R}^{m\times 1}$ is the vector of the model parameters to be trained, λ, $C_p = \frac{1}{n_p}c_p$, $C_u = \frac{1}{n_u}c_u$, as defined in Table 1.1, are positive hyper-parameters, $[g(\cdot)]_+ = \max(0, g(\cdot)) \in \mathbb{R}^{n_p\times 1}$, $\boldsymbol{R}$ is the similarity matrix obtained by Equation 2.6, and $\mathbf{1}_{p,u,pu} = (\underbrace{1,1,\cdots,1}_{k})^T, k = n_p, n_u, n_p + n_u$, which is formulated as Equation 2.7.

The predictive score function of GLPUAL for instance $\boldsymbol{x}$ is the same as the predictive score function of SVM, i.e.,

$$f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0.$$

3.3 Algorithm of GLPUAL for Linear Decision Boundary in the Original Feature Space

3.3.1 Alternating Direction Method of Multipliers (ADMM)

The hinge loss term $\mathbf{1}_p^T\left[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)\right]_+$ in the objective function of GLPUAL in Equation 3.2 is not always differentiable in the feasible region of the optimisation in Equation 3.2, bringing difficulty to applying the gradient descent directly. To find an alternative way for solving the GLPUAL, the following reformulation of the optimisation of GLPUAL in Equation 3.2 can be considered:

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0,\boldsymbol{h}} & C_p \mathbf{1}_p^T[\boldsymbol{h}]_+ + C_u(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)^T(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0) \\ & + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0) + \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} \\ & s.t.\ \boldsymbol{h} = \mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0).\end{aligned} \tag{3.3}$$

The convex objective function in Equation 3.3 can be regarded as the sum of the functions of $(\boldsymbol{\beta},\beta_0)$ and the function of $\boldsymbol{h}$ while the constraints in Equation 3.3 can be regarded as a linear combination of $(\boldsymbol{\beta},\beta_0)$ and $\boldsymbol{h}$; this meets the requirement of ADMM, which was proposed in [37] to decompose a large-scale convex optimisation problem with affine constraints into several simpler sub-problems and update the solution iteratively until convergence. ADMM was introduced to the field of machine learning by [38]. Moreover, ADMM is able to converge to modest accuracy within fewer iterations than the gradient descent.

The Lagrangian function of problem in Equation 3.3 is

$$\begin{aligned}\mathscr{L}(\boldsymbol{\theta}) =& C_p \mathbf{1}_p^T[\boldsymbol{h}]_+ + C_u(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)^T(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0) \\ & + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0) + \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} \\ & + \boldsymbol{u}_{\boldsymbol{h}}^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}],\end{aligned} \tag{3.4}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\beta},\beta_0,\boldsymbol{h},\boldsymbol{u_h}\}$ and $\boldsymbol{u_h}$ is dual variable. Then the augmented Lagrangian function is given as

$$\mathscr{L}_a(\boldsymbol{\theta}) = \mathscr{L}(\boldsymbol{\theta}) + \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}\right\|_2^2. \tag{3.5}$$

According to the ADMM, the optimal solution of the GLPUAL can be found by solving the following updates of iteration until convergence:

$$\begin{aligned}
(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)}) &= \arg\min_{\boldsymbol{\beta},\beta_0} \mathscr{L}_a(\boldsymbol{\beta},\beta_0,\boldsymbol{h}^{(k)},\boldsymbol{u}_{\boldsymbol{h}}^{(k)}),\\
\boldsymbol{h}^{(k+1)} &= \arg\min_{\boldsymbol{h}} \mathscr{L}_a(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)},\boldsymbol{h},\boldsymbol{u}_{\boldsymbol{h}}^{(k)}),\\
\boldsymbol{u}_{\boldsymbol{h}}^{(k+1)} &= \boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}].
\end{aligned} \tag{3.6}$$

3.3.2 Update of β and β_0

The update of $\boldsymbol{\beta}$ and β_0 is

$$\begin{aligned}
(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)}) = \arg\min_{\boldsymbol{\beta},\beta_0} &\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\\
&+ C_u(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)^T(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)\\
&+ (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)\\
&+ \boldsymbol{u}_{\boldsymbol{h}}^{(k)^T}[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}]\\
&+ \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}\right\|_2^2,
\end{aligned} \tag{3.7}$$

which is a quadratic optimisation with every term differentiable.

Let $\boldsymbol{I}_k, \forall k \in \mathbb{Z}$ denote a $k \times k$ identity matrix and by defining

$$\begin{aligned}
\boldsymbol{M}_{11} &= \lambda\boldsymbol{I}_m + 2C_u\boldsymbol{X}_{[u]}^T\boldsymbol{X}_{[u]} + 2\boldsymbol{X}_{[pu]}^T\boldsymbol{R}\boldsymbol{X}_{[pu]} + \mu_1\boldsymbol{X}_{[p]}^T\boldsymbol{X}_{[p]},\\
\boldsymbol{M}_{12} &= 2C_u\boldsymbol{X}_{[u]}^T\mathbf{1}_u + 2\boldsymbol{X}_{[pu]}^T\boldsymbol{R}\mathbf{1}_{pu} + \mu_1\boldsymbol{X}_{[p]}^T\mathbf{1}_p,\\
\boldsymbol{M}_{21} &= 2C_u\mathbf{1}_u^T\boldsymbol{X}_{[u]} + 2\mathbf{1}_{pu}^T\boldsymbol{R}\boldsymbol{X}_{[pu]} + \mu_1\mathbf{1}_p^T\boldsymbol{X}_{[p]},\\
M_{22} &= 2C_u n_u + 2\mathbf{1}_{pu}^T\boldsymbol{R}\mathbf{1}_{pu} + \mu_1 n_p,\\
\boldsymbol{m}_1 &= -2C_u\boldsymbol{X}_{[u]}^T\mathbf{1}_u + \boldsymbol{X}_{[p]}^T\boldsymbol{u}_{\boldsymbol{h}} + \mu_1\boldsymbol{X}_{[p]}^T(\mathbf{1}_p - \boldsymbol{h}),\\
m_2 &= -2C_u n_u + \boldsymbol{u}_{\boldsymbol{h}}^T\mathbf{1}_p + \mu_1(\mathbf{1}_p - \boldsymbol{h})^T\mathbf{1}_p,
\end{aligned} \tag{3.8}$$

the solution of problem in Equation 3.7 can be obtained by solving the following linear equation w.r.t. $\boldsymbol{\beta}$ and β_0:

$$\begin{bmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & M_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}^{(k+1)}\\ \beta_0^{(k+1)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{m}_1\\ m_2\end{bmatrix}. \tag{3.9}$$

3.3.3 Update of h

The update of $\boldsymbol{h}$ is

$$\begin{aligned}\boldsymbol{h}^{(k+1)} =& \arg\min_{\boldsymbol{h}} C_p \mathbf{1}_p^T [\boldsymbol{h}]_+ + {\boldsymbol{u}_{\boldsymbol{h}}^{(k)}}^T [\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \boldsymbol{h}] \\ &+ \frac{\mu_1}{2} \left\| \mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \boldsymbol{h} \right\|_2^2,\end{aligned} \tag{3.10}$$

which is equivalent to solving the problem

$$\min_{\boldsymbol{h}} \sum_{i=1}^{n_p} \left\{ \frac{C_p}{\mu_1}[h_i]_+ + \frac{1}{2}[1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{x}_i^T \boldsymbol{\beta}^{(k+1)} + \beta_0^{(k+1)}) - h_i]^2 \right\}. \tag{3.11}$$

According to [39], for constant $c > 0$, we can obtain

$$\arg\min_x c[x]_+ + \frac{1}{2}\|x - d\|_2^2 = \begin{cases} d - c, d > c, \\ 0, 0 \le d \le c, \\ d, d < 0. \end{cases} \tag{3.12}$$

Thus, by defining $s_c(d) = \arg\min_x c[x]_+ + \frac{1}{2}\|x-d\|_2^2$, the ith element of $\boldsymbol{h}^{(k+1)}$ in problem in Equation 3.10 is solved as

$$h_i^{(k+1)} = s_{\frac{C_p}{\mu_1}}\left(1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{x}_i^T \boldsymbol{\beta}^{(k+1)} + \beta_0^{(k+1)})\right), i = 1, 2, \ldots, n_p. \tag{3.13}$$

3.4 Experiments on Synthetic Data

In this section, we conducted experiments on synthetic datasets following the pattern in Figure 3.2 to access the performance of GLPUAL compared with GLLC.

3.4.1 The Generation of Synthetic Positive-Negative (PN) Datasets

The simple synthetic datasets following the pattern in Figure 3.2 were obtained by the following steps:

1. To generate the first subset of the 2-dimensional synthetic positive set, 200 instances were sampled from the multivariate normal distribution with mean vector $(15,15)$ and the covariance matrix
$$\begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}.$$
2. To generate the second subset of the 2-dimensional synthetic positive set, 200 instances were sampled from the multivariate normal distribution with the mean vector $(\mathbf{mean}_{p2}, \mathbf{mean}_{p2})$ and the covariance matrix
$$\begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}.$$
3. To generate the 2-dimensional synthetic negative set, 400 instances were sampled from the multivariate normal distribution of mean vector $(0,0)$ and the covariance matrix
$$\begin{bmatrix} 50 & 0.2 \\ 0.2 & 50 \end{bmatrix}.$$
4. Mixing the first subset of the synthetic positive set obtained in Step 1, the second subset of the synthetic positive set obtained in Step 2 and the synthetic negative set obtained in Step 3, simple 2-dimensional synthetic dataset can be eventually obtained as shown in Figure 3.2.

In Step 2, $\mathbf{mean}_{p2}$ took value from $(50, 100, 200, 500, 1000)$. For each value of $\mathbf{mean}_{p2}$, the above steps were repeated 5 times so that we have 5 synthetic datasets for each value of $\mathbf{mean}_{p2}$.

3.4.2 Training-Test Split for the Synthetic PU Datasets

It should be noted that both GLLC and GLPUAL can be applied on the datasets sampled from either single-training-set scenario or case-control scenario as we set the suitable metric for hyper-parameter tuning in practice. More specifically, the case-control scenario indicates that the unlabeled training set can be regarded to be i.i.d. sampled from the population, while the single-training-set scenario indicates that the whole training set can be regarded to be i.i.d. sampled from the population. In this case, for more intuitive comparison, we split each of the synthetic dataset generated in Section 3.4.1 to construct the PU training and test sets consistent with the single-training-set scenario by the following two steps:

1. Firstly, to split the dataset into a the training set and a test set, 70% of the instances in the simple synthetic dataset obtained in Section 3.4.1 were randomly selected as the training set while the test set was constituted by the rest 30% instances.

2. Secondly, to construct the labeled-positive set and unlabeled-set for training, 25% of the positive instances in the above obtained training set were randomly selected to form the labeled-positive set for training. The rest of the positive instances were mixed with the negative set, contributing to the unlabeled set for training.

Then 25 pairs of PU training set and test set were obtained.

3.4.3 Model Setting

It should be noted that the real value of the F1-score on the training dataset is not accessible if we do not use the label information during model training. Therefore, by fixing C_p to 1 and the number K of the nearest neighbors to 5, C_u, λ, σ, in the objective functions of GLPUAL and GLLC, were determined by the 4-fold cross-validation (CV), which reached the highest average PUF-score proposed in [40] on the validation sets. PUF-score is similar to F1-score and can be directly obtained from PU data, i.e.,

$$\text{PUF-score} = \frac{\text{recall}^2}{P[\operatorname{sgn}(f(\boldsymbol{x})) = 1]}, \tag{3.14}$$

where recall can be estimated by computing $\frac{1}{n_p}\sum_{\boldsymbol{x}_i \in p} \beth(\operatorname{sgn}(f(\boldsymbol{x}_i)) = 1)$ with the indicator function denoted by $\beth(\cdot)$, and at the single-training-set scenario $P[\operatorname{sgn}(f(\boldsymbol{x})) = 1]$ can be estimated by computing $\frac{1}{n_p+n_u}\sum_{\boldsymbol{x}_i \in u \cup p} \beth(\operatorname{sgn}(f(\boldsymbol{x}_i)) = 1)$. Furthermore, λ, σ were tuned from the set $\{1,2,3,4,5\} \circ \{0.1,1,10,100\}$. C_u was selected to from the set $\{0.01, 0.02, \ldots, 0.5\}$ based on the setting in [21].

3.4.4 Results and Analysis

The results of the experiments, on the constructed synthetic PU datasets are summarised in Table 3.1. The results are measured by the average F1-score, which is a popular metric for the evaluation of PU learning methods [41]. The patterns of the decision boundary obtained by GLPUAL and GLLC on the synthetic datasets are illustrated in Figure 3.4. For each value of $\mathbf{mean}_{p2}$, we use the first generated synthetic dataset as example.

Table 3.1: Summary of the average F1-score (%) of the experiments on the synthetic dataset with the standard deviations.

$\mathbf{mean}_{p2}$	GLPUAL	GLLC
50	95.07 ± 0.78	91.17 ± 1.52
100	94.89 ± 0.61	86.27 ± 1.83
200	93.56 ± 0.75	81.04 ± 2.93
500	92.83 ± 3.22	73.58 ± 2.58
1000	93.47 ± 2.08	71.28 ± 2.37

According to the experimental results in Table 3.1, GLPUAL always has better performance than GLLC on the synthetic PU datasets with all the 5 values of $\mathbf{mean}_{p2}$. Furthermore, with the value of $\mathbf{mean}_{p2}$ increasing, the gap between GLPUAL and GLLC becomes increasingly large. This is more clearly in the ten plots in Figure 3.4, as one of the positive subset becomes increasingly far away, the decision boundary of GLLC is stretched to a strange position so that there are more and more instances misclassified by the decision boundary of GLLC, while the decision boundary of

GLPUAL is unaffected. Therefore it is verified that GLPUAL can have better performance to generate the linear decision boundary than GLLC on the datasets following the pattern in Figure 3.2.

3.5 Kernel Trick to GLPUAL for Non-Linear Decision in the Original Feature Space

Now we need to introduce the kernel trick to GLPUAL for the non-linear decision boundary in the original space so that to make GLLC able to be applied on the non-linear separable datasets including trifurcated datasets. The techniques applied in this section are similar to many previous methods [21, 42, 43, 44].

Suppose $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{r\times 1}$ be a mapping of the instance vector $\boldsymbol{x}$. Then let $\boldsymbol{\phi}(\boldsymbol{X}_{[k]}) \in \mathbb{R}^{n_k \times r}, k = p, u, pu$ be the mapping of the original data matrix $\boldsymbol{X}_{[k]}$. The ith row of $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})$ is $\phi(\boldsymbol{x}_i)^T$. According to Equations 3.8 and 3.9, using $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})$ as features matrix instead of $\boldsymbol{X}_{[k]}$ for the training of GLPUAL, we can find the following necessary condition for the optimal solution of $\boldsymbol{\beta}$:

$$\boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T \boldsymbol{\Omega}, \tag{3.15}$$

where

$$\boldsymbol{B} = \boldsymbol{M}_{11} - \frac{\boldsymbol{M}_{12}\boldsymbol{M}_{21}}{M_{22}}, \tag{3.16}$$

and

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{u_h} - \mu_1 \frac{m_2}{M_{22}} \mathbf{1}_p + \mu_1 (\mathbf{1}_p - \boldsymbol{h}) \\ -2C_u \mathbf{1}_u - 2\frac{m_2}{M_{22}} C_u \mathbf{1}_u \end{bmatrix} - 2\frac{m_2}{M_{22}} \boldsymbol{R}\mathbf{1}_{pu}. \tag{3.17}$$

Equation 3.15 can be regarded as a condition when the objective function reaches its minimum. Since $\boldsymbol{B}$ is symmetric and invertible, we can obtain

$$\boldsymbol{\beta} = \boldsymbol{B}^{-1} \boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T \boldsymbol{\Omega}. \tag{3.18}$$

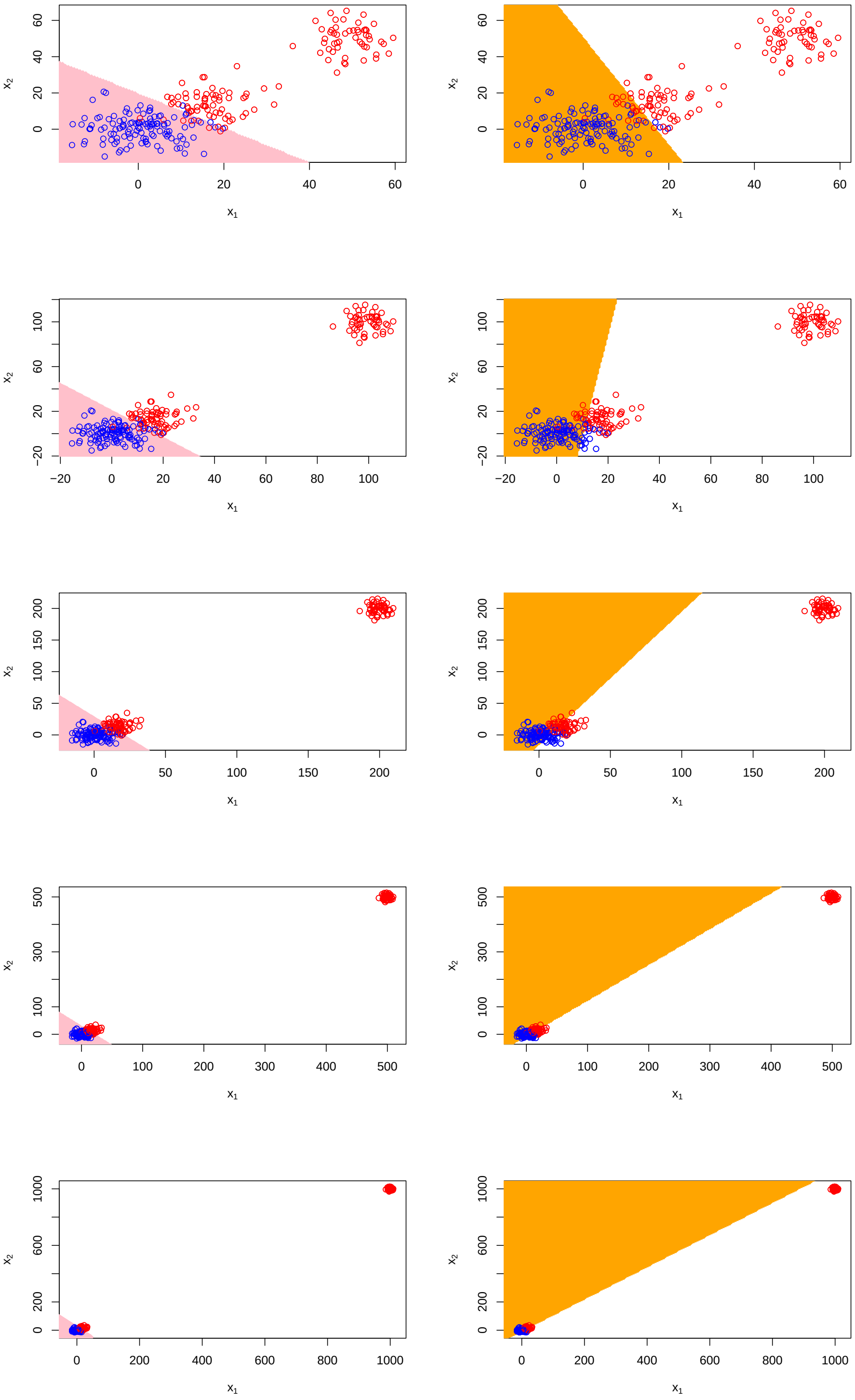

Figure 3.4: The decision boundaries trained by GLPUAL and GLLC on the synthetic data with $\mathbf{mean}_{p2} = 50, 100, 200, 500, 1000$; pink area: the negative field of GLPUAL; orange area: the negative field of GLLC; the instances in the plots are from the test sets; red: positive instances; blue: negative instances.

Substituting Equation 3.18 into the objective function in Equation 3.2, we have

$$\begin{aligned}\min_{\boldsymbol{\Omega},\beta_0}\frac{\lambda}{2}\boldsymbol{\Omega}^T\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+C_p\mathbf{1}_p^T[\mathbf{1}_p-(\boldsymbol{\phi}(\boldsymbol{X}_{[p]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+\mathbf{1}_p\beta_0)]_+\\ +C_u[\mathbf{1}_u+\boldsymbol{\phi}(\boldsymbol{X}_{[u]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+\beta_0\mathbf{1}_u][\mathbf{1}_u+\boldsymbol{\phi}(\boldsymbol{X}_{[u]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+\beta_0\mathbf{1}_u]^T\\ +(\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}+\mathbf{1}_{pu}\beta_0).\end{aligned} \tag{3.19}$$

To prove $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T$ and $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T$ are kernel matrices for $\boldsymbol{X}_{[k]}$ and $\boldsymbol{X}_{[pu]}$ in Equation 3.19, we need to introduce two properties for the construction of kernel functions proved in [45] as

Theorem 1 *Let $\phi(\boldsymbol{X}),\phi(\boldsymbol{Z})$ be a mapping of matrices of $\boldsymbol{X},\boldsymbol{Z}$ and $\boldsymbol{\kappa}_1(\phi(\boldsymbol{X}),\phi(\boldsymbol{Z}))$ be a kernel matrix of $\phi(\boldsymbol{X})$ and $\phi(\boldsymbol{Z})$. Then the following two matrices $\boldsymbol{\kappa}_2(\boldsymbol{X},\boldsymbol{Z})$ and $\boldsymbol{\kappa}_3(\boldsymbol{X},\boldsymbol{Z})$ can be regarded as the kernel matrix w.r.t. $\boldsymbol{X},\boldsymbol{Z}$:*

- $\boldsymbol{\kappa}_2(\boldsymbol{X},\boldsymbol{Z})=\boldsymbol{\kappa}_1(\phi(\boldsymbol{X}),\phi(\boldsymbol{Z}))$,
- $\boldsymbol{\kappa}_3(\boldsymbol{X},\boldsymbol{Z})=\boldsymbol{X}\boldsymbol{F}\boldsymbol{Z}^T$*, with $\boldsymbol{F}$ to be a symmetric matrix.*

According to the closure properties in Theorem 1, we can obtain

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T=&\boldsymbol{\Phi}'(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))\\ =&\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})\end{aligned} \tag{3.20}$$

and

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T=&\boldsymbol{\Phi}''(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))\\ =&\boldsymbol{\Phi}_2(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]}),\end{aligned} \tag{3.21}$$

where $\boldsymbol{\Phi}'(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))$, $\boldsymbol{\Phi}''(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))$ are the kernel matrices for $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})$ and $\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})$, and $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]}),\boldsymbol{\Phi}_2(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})$ are the kernel matrices for $\boldsymbol{X}_{[k]}$ and $\boldsymbol{X}_{[pu]}$.

Thus the objective function of GLPUAL can be reformulated as

$$\begin{aligned}
&\min_{\boldsymbol{\Omega},\beta_0} \frac{\lambda}{2}\boldsymbol{\Omega}^T \Phi_2(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + C_p \mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{\Phi}(\boldsymbol{X}_{[p]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_p\beta_0)]_+ \\
&+ C_u[\mathbf{1}_u + \boldsymbol{\Phi}(\boldsymbol{X}_{[u]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \beta_0\mathbf{1}_u]^T[\mathbf{1}_u + \boldsymbol{\Phi}(\boldsymbol{X}_{[u]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \beta_0\mathbf{1}_u] \\
&+ (\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0),
\end{aligned} \tag{3.22}$$

whose solution is only determined by the kernels.

The predictive score function for instance $\boldsymbol{x}^*$ of GLPUAL can be transformed to

$$f = \boldsymbol{\Phi}(\boldsymbol{x}^*,\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \beta_0. \tag{3.23}$$

In this case, we can update β_0 via

$$\beta_0^{(k+1)} = \frac{m_2}{M_{22}} - \boldsymbol{Q}_b^{(k+1)}/M_{22}, \tag{3.24}$$

where m_2, M_{22} are not related to $\boldsymbol{X}_{[p]},\boldsymbol{X}_{[u]},\boldsymbol{X}_{[pu]}$ and

$$\begin{aligned}
\boldsymbol{Q}_b^{(k+1)} =& 2C_u\mathbf{1}_u^T\boldsymbol{\Phi}(\boldsymbol{X}_{[u]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} \\
&+ 2\mathbf{1}_{pu}^T\boldsymbol{R}\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} \\
&+ \mu_1\mathbf{1}_p^T\boldsymbol{\Phi}(\boldsymbol{X}_{[p]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)}.
\end{aligned} \tag{3.25}$$

The update of $\boldsymbol{h},\boldsymbol{u}_{\boldsymbol{h}}$ can be reformulated as

$$\begin{aligned}
\boldsymbol{h}_i^{(k+1)} &= s_{\frac{C_p}{\mu_1}}\left(1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{\Phi}(\boldsymbol{x}_i,\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} + \beta_0^{(k+1)})\right), i = 1,2,\ldots,n_p, \\
\boldsymbol{u}_{\boldsymbol{h}}^{(k+1)} &= \boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{\Phi}(\boldsymbol{X}_{[p]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}].
\end{aligned} \tag{3.26}$$

Thus the update of ADMM for non-linear decision boundary can be summarised into the following steps:

1. Set initial values of $\boldsymbol{\Omega}$, β_0, $\boldsymbol{h}$, $\boldsymbol{u_h}$.

2. Update $\boldsymbol{\Omega}$ via Equation 3.17 w.r.t. $\boldsymbol{h}^{(k)}$ and $\boldsymbol{u}_{\boldsymbol{h}}^{(k)}$.

3. Update β_0 via Equation 3.24.

4. Update $\boldsymbol{h}$,$\boldsymbol{u_h}$ via Equation 3.26.

5. Repeat Step 2 and Step 3 until convergence.

$\boldsymbol{\Phi}_2(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})$ does not directly appear in the update process for the optimisation in this section so that we only need to determine the form of $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})$ in practice. Moreover, λ either does not appear directly in the above update process or it is contained in the matrix $\boldsymbol{B}$ as a part of $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})$. Therefore, for convenience, in the case of using the kernel trick in Section 3.4, we use λ to represent the hyper-parameter(s) of the kernel matrix $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})$.

3.6 Experiments on Real Data

In this section, for the further explore the performance of GLPUAL to verify our motivation, the experiments were conducted on real datasets.

3.6.1 The Source of Real Datasets

Firstly, 16 datasets from the UCI Machine Learning Repository[1] were selected to assess the performance of GLPUAL and verify our motivation: Accelerometer (**Acc**), **Ecoli**, Pen-Based Recognition of Handwritten Digits (**Pen**), Online Retail (**OR1**), Online Retail II (**OR2**), Parking Birmingham (**PB**), Wireless Indoor Localization (**wifi**), Sepsis survival minimal clinical records (**SSMCR**), Avila, Raisin Dataset (**RD**), Occupancy Detection (**OD**), User Knowledge Modeling Data Set (**UMD**), **Seeds**, Energy efficiency Data Set (**ENB**), Heart Disease (**HD**) and Liver Disorders (**LD**). The details of these 16 real datasets are summarised in Table 3.2.

[1] UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/index.php`

Table 3.2: Summary of the datasets for the verification of the motivation of GLPUAL.

Dataset	positive instances	negative instances	features
Acc	100 red	100 blue	4
Ecoli	116 im & 52 pp	143 cp & 25 om	6
Pen	200 one & 200 eight	400 four	16
OR1	301 UK	301 Germany	4
OR2	500 UK	500 Germany	4
SSMCR	391 alive	109 dead	3
PB	500 Bull Ring	500 BHMBCCMKT01	3
OD	100 occupied	300 not occupied	5
UMD	83 Low	63 high	5
Seeds	70 Kama	70 Rosa	7
ENB	144 TypeII	144 Type III	7
wifi	100 Location 2& 100 Location 4	499 Location 1 & 100 Location 3	7
Avila	300 E	900 A	10
RD	450 Kecimen	450 Besni	7
LD	144 class 1	200 class 2	6
HD	150 absence	119 presence	13

3.6.2 Training-Test Split for the Real PU Datasets

Different from the steps in Section 3.4.2, PU training and test sets for the 16 real datasets are constructed under the case-control scenario since we would like to see the performance of GLPUAL on the datasets with various labeling mechanism. Furthermore, under the case-control scenario, it is possible to do fair comparison between GLLC and the two convincing methods for PU classification, i.e., uPU [23] and nnPU [24], since they were proposed under the case-control scenario. The steps to do training-test split for the 16 PU real datasets are summarised as follows:

1. To obtain the binary positive-negative (PN) datasets from the original multi-class dataset, certain classes in each of the original real datasets were treated as positive while some other classes were treated as negative with the rest of classes abandoned.

2. To construct the labeled-positive set and unlabeled-set, γ' of the positive instances in each binary PN dataset obtained in Step 1 were randomly selected to form the labeled-positive set, and the rest of the positive instances were mixed into negative set, contributing to the unlabeled set of the PU dataset.

3. To generate the training set and the test set from the constructed PU datasets, the labeled-positive set and 70% of the instances in the unlabeled set were selected as the training set while the test set was constituted by the rest 30% of the instances in the unlabeled set; this is corresponding to the setting of case-control scenario since the unlabeled training set and the test set can be regarded to be sampled from the same population.

During the preliminary experiment, we found that the label frequency γ needs to be greater than 0.2 for the hyper-parameter tuning strategy introduced in Section 3.6.4 to achieve satisfactory results. Therefore, the value of γ' is set to $\frac{7}{17}$, $\frac{7}{37}$ so that we have the label frequency $\gamma = \gamma'/(0.3\gamma' + 0.7) = 0.5, 0.25$, which is the fraction of positive instances that are labeled, in the corresponding constructed PU training sets, respectively. Then Step 2 and Step 3 were repeated for 10 times on each of the 16 binary PN datasets and obtained 10 pairs of PU training and test sets for each of the 16 binary PN datasets with a certain value of γ'.

3.6.3 Compared Methods

GLLC, uPU and nnPU were also trained on the 16 real datasets as the compared methods with GLPUAL. GLLC serves as the baseline of GLPUAL. uPU and nnPU are not only two convincing methods for PU classification as mentioned in Section 3.6.2, but also they are related to the subsequent works of GLPUAL in this study.

3.6.4 Model Setting

At the case-control scenario, we also use PUF-score in Equation 3.14 for hyper-parameter selection. The numerator 'recall' can still be estimated by computing $\frac{1}{n_p}\sum_{\boldsymbol{x}_i \in p} \daleth(\mathrm{sgn}(f(\boldsymbol{x}_i)) = 1)$, while the denominator $P[\mathrm{sgn}(f(\boldsymbol{x})) = 1]$ needs to be estimated by computing $\frac{1}{n_u}\sum_{\boldsymbol{x}_i \in u} \daleth(\mathrm{sgn}(f(\boldsymbol{x}_i)) = 1)$ at the case-control scenario. Therefore, by fixing C_p to 1 and the number K of the nearest neighbors to 5, C_u, λ, σ in the objective functions of GLPUAL and GLLC were firstly tuned by 4-fold CV, which reached the highest average PUF-score.

Furthermore, considering the higher complexity of the real datasets compared with the synthetic datasets in Section 3.4, we modified our strategy for hyper-parameter selection, enabling efficient selection of hyper-parameters across a broader range. More specifically, λ, σ were tuned from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$ and C_u was selected to from the set $\{0.5, 0.3, 0.1, 0.05, 0.01\}$ based on the setting in [21]. Then λ, σ and C_u were continually tuned following the greedy algorithm based on the average PUF-score on the validation sets as follows:

1. Set λ, σ and C_u to the best combination from the grid search.

2. Sequentially update one of hyper-parameters λ, σ and C_u by increasing/decreasing 10% of its current value with the rest of the hyper-parameters unchanged. The optimal scenario on 4-fold CV is set to be the final update of this step.

3. Repeat Step 2 until there is no better scenario appeared.

Besides, the hyper-parameters of uPU and nnPU were fixed as the recommended setting in the open source provided by [24] on GitHub. Radial Basis Function (RBF) kernel was applied on both GLPUAL and GLLC. More specifically, we computed $\exp\left(-\|x_i - x_j\|^2/2\lambda^2\right)$ as the (i, j) element of $\boldsymbol{\Phi}'(\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}), \boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))$.

3.6.5 Summary of Results

The performance of GLPUAL, GLLC, uPU and nnPU on the 16 real datasets are summarised in Table 3.3.

The results of the difference between the F1-score of GLPUAL and GLLC on each pairwise experiments are shown in the boxplots in Figure 3.5, Figure 3.6 with $\gamma = 0.5, 0.25$, respectively. In all of these figures, the best four datasets for GLPUAL compared with GLLC are **wifi**, **OR1**, **OR2**, and **Pen**. Furthermore, with $\gamma = 0.5, 0.25$, three of the worst four datasets for GLPUAL compared with GLLC are **OD**, **LD**, **Avila**. When $\gamma = 0.5$, the rest one of the worst four datasets is **PB**. When $\gamma = 0.25$, the rest one of the worst four datasets is **RD**.

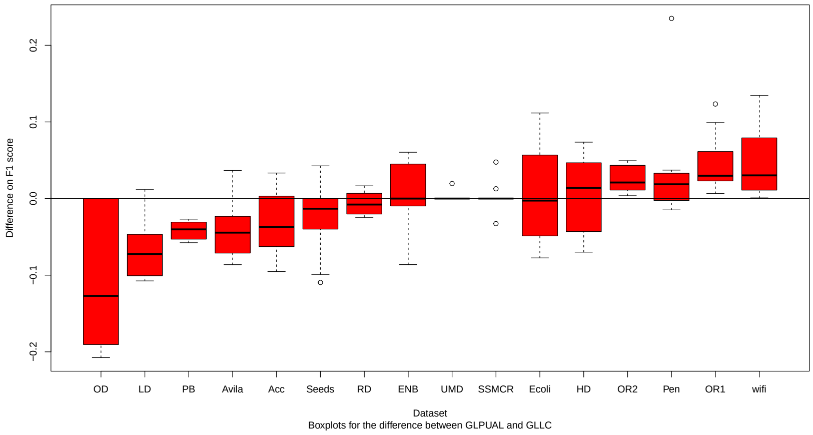

Figure 3.5: Boxplots for the difference between F1-scores of GLPUAL and GLLC on each dataset increasingly ranked by medians; label frequency γ=0.5; x-axis: the datasets; y-axis: the difference between GLPUAL and GLLC on F1-scores.

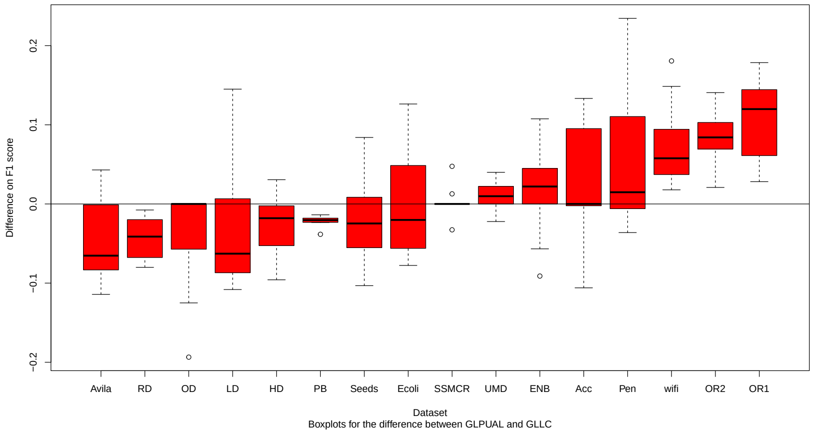

Figure 3.6: Label frequency γ=0.25. The rest of the caption is as in Figure 3.5.

3.6.6 Pattern Analysis for the Real Datasets Preferring GLPUAL to GLLC

The t-SNE plots of the best four datasets, i.e., **wifi**, **OR1**, **OR2**, and **Pen**, for GLPUAL compared with GLLC are shown in Figure 3.7. According to the four t-SNE plots, the following information can be summarised:

1. The trifurcated pattern of the best four datasets in Figure 3.7 is clear that the positive set is constituted by two subsets distributing on both sides of the negative set as discussed in the motivation of GLPUAL in Section 3.1.

2. In the t-SNE plot of dataset **OR1**, there are much more instances contained in the right labeled-positive subset than the instances contained in the left labeled-positive set. This indicates that the trifurcated pattern does not have to be balanced for GLPUAL to outperform GLLC.

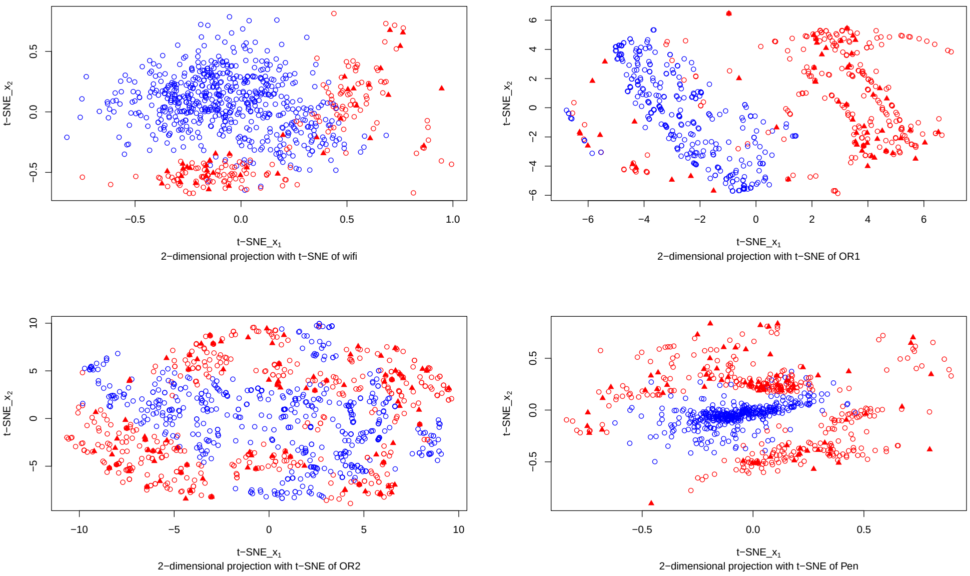

Figure 3.7: The t-SNE plots of the best four datasets **wifi**, **OR1**, **OR2** and **Pen**; cross entropy loss for training with perplexity = 750, 200, 250, 50; label frequency γ=0.25; red: positive instances; blue: negative instances; triangle: labeled instances; circle: unlabeled instances.

3.6.7 Pattern Analysis for the Real Datasets Preferring GLLC to GLPUAL

There are overall five datasets to be the worst four datasets for GLPUAL compared with GLLC under two cases of label frequency, i.e., $\gamma = 0.5, 0.25$. The t-SNE plots for the six datasets are illustrated in Figure 3.8 and their patterns can be summarised as follows:

1. According to the t-SNE plots of the two datasets **LD** and **Avila**, the positive set and the negative set are mixed together, making the dataset challenging to be separated. In this case, the labeled-positive instances selected as support vectors by the hinge loss of GLPUAL are not sufficient to adequately represent

the pattern of the positive set, while the squared loss of GLLC on the labeled-positive set, which selects all positive instances as the support vectors, can somewhat alleviate this issue. As a result, GLLC on these two datasets outperforms GLPUAL.

2. The t-SNE plots of the four datasets **OD**, **PB**, and **RD** represent the typical two-class patterns. In this type of datasets, the problem of GLLC mentioned in Section 3.1 does not exist. Therefore, under the strategy of hyper-parameter tuning in Section 3.6.4, the optimal combination(s) of the hyper-parameters for GLLC to outperform GLPUAL was (were) found in at least one case of label frequency γ.

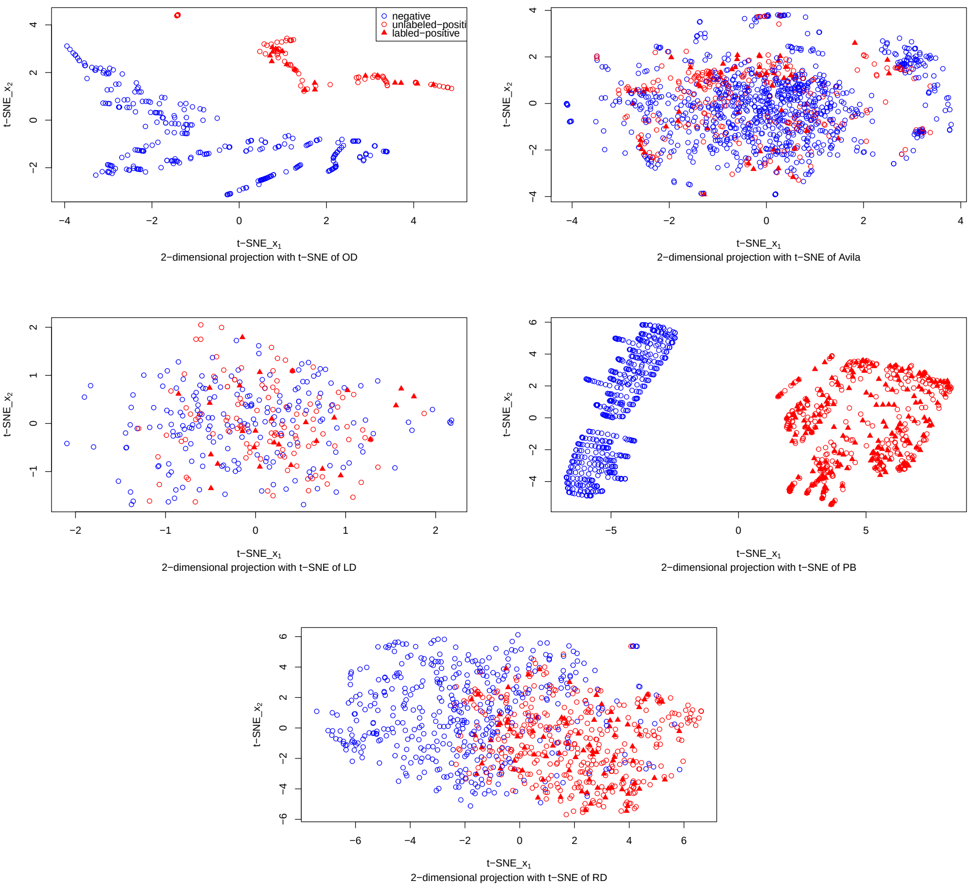

Figure 3.8: The t-SNE plots of the worst six datasets **OD**, **Avila**, **LD**, **PB**, **RD**, **Ecoli**; cross entropy loss for training with perplexity = 200, 550, 250, 300, 300; label frequency γ=0.25, 0.25, 0.25, 0.5, 0.25. The rest of the caption is as in Figure 3.7.

3.6.8 Results among GLPUAL, uPU and nnPU

According to the results in Table 3.3, there are in total 22 cases of the 32 cases where GLPUAL outperformed uPU and nnPU. Furthermore, uPU and nnPU sometimes have much larger standard deviations than GLPUAL since their algorithms based on ADAM for the optimisation of their non-convex objective functions cannot always converge to the optimal solution, though nnPU alleviated this issue to some extent. Finally, there are 11 cases where CPB-GLPUAL is the optimal choice among the four methods in the experiments.

Table 3.3: The average F1-score (%) of the classifiers; for each of the 16 original datasets, the average F1-score (%) and standard deviation in the two rows were obtained under label frequency $\gamma = 0.5, 0.25$, respectively; the results highlighted in blue for GLPUAL indicate that it is the best among the four methods.

Dataset	GLPUAL	GLLC	uPU	nnPU
ENB	42.82 ± 4.76	42.69 ± 4.62	29.58 ± 22.14	30.20 ± 23.67
	45.82 ± 7.50	44.16 ± 6.56	26.12 ± 30.53	26.88 ± 31.28
HD	82.72 ± 2.35	81.97 ± 5.45	71.38 ± 4.23	74.38 ± 2.19
	81.92 ± 4.03	84.46 ± 4.11	71.01 ± 3.97	75.06 ± 2.40
Pen	92.47 ± 8.13	88.92 ± 10.15	77.76 ± 31.00	87.50 ± 14.94
	91.73 ± 9.04	87.02 ± 11.38	72.55 ± 31.03	84.06 ± 16.85
LD	44.24 ± 5.72	50.79 ± 6.86	11.88 ± 25.75	31.54 ± 27.79
	36.85 ± 9.97	40.05 ± 8.85	10.15 ± 22.39	20.09 ± 26.27
OR1	90.05 ± 2.25	85.62 ± 3.77	16.64 ± 33.33	84.08 ± 6.88
	83.88 ± 5.78	72.95 ± 5.50	20.90 ± 33.12	72.06 ± 6.99
RD	82.45 ± 2.18	83.01 ± 2.38	70.61 ± 12.89	71.29 ± 13.63
	77.63 ± 3.77	81.99 ± 2.84	72.92 ± 14.53	73.12 ± 12.83
Seeds	92.31 ± 4.86	94.63 ± 2.81	92.37 ± 1.51	97.25 ± 3.65
	89.05 ± 5.53	91.18 ± 4.48	86.85 ± 3.12	93.08 ± 3.89
wifi	95.10 ± 1.96	90.43 ± 4.65	91.16 ± 4.29	92.17 ± 3.17
	96.69 ± 1.83	89.09 ± 4.77	87.69 ± 2.88	89.27 ± 2.61
Avila	55.82 ± 2.90	59.56 ± 4.10	62.74 ± 8.82	63.75 ±9.07
	50.05 ± 4.22	54.99 ± 6.17	61.30 ± 9.23	61.00 ± 9.04
OD	89.00 ± 8.44	100.00 ± 0.00	80.00 ± 42.16	100.00 ± 0.00
	95.69 ± 6.74	100.00 ± 0.00	80.00 ± 42.16	100.00 ± 0.00
OR2	88.93 ± 1.22	86.49 ± 1.38	76.92 ± 4.90	81.60 ± 4.23
	85.50 ± 3.42	77.10 ± 5.65	74.41 ± 5.45	77.28 ± 3.76
PB	95.90 ± 1.12	100.00 ± 0.00	69.77 ± 2.62	67.19 ± 3.17
	97.86 ± 0.67	100.00 ± 0.00	68.75 ± 2.63	66.63 ± 4.09

Table 3.3 – continued from previous page

Dataset	GLPUAL	GLLC	uPU	nnPU
Acc	65.02 ± 4.79 66.36 ± 4.42	68.10 ± 2.18 64.08 ± 6.31	20.05 ± 27.57 21.95 ± 29.61	20.46 ± 28.56 23.43 ± 31.40
Ecoli	90.80 ± 2.59 88.04 ± 4.44	90.15 ± 6.28 88.03 ± 3.96	84.41 ± 6.13 84.92 ± 6.82	85.92 ± 6.69 86.05 ± 6.55
SSMCR	87.63 ± 1.29 87.63 ± 1.29	87.35 ± 2.14 87.35 ± 2.14	85.71 ± 1.98 84.97 ± 2.03	87.42 ± 1.37 86.79 ± 1.50
UMD	100.00 ± 0.00 99.58 ± 0.88	99.80 ± 0.62 98.41 ± 1.80	100.00 ± 0.00 100.00 ± 0.00	100.00 ± 0.00 100.00 ± 0.00

3.7 Conclusion

In this chapter, firstly GLPUAL was proposed for better classification on the datasets, where the distances between the two positive subsets to the ideal decision boundary are very different. Secondly, an algorithm to solve the optimisation of GLPUAL was proposed based on ADMM with linear decision boundary generated. Then experiments were conducted on the synthetic datasets to verify the motivation of the proposed method. Thirdly, the kernel trick was introduced to GLPUAL and then the algorithm to solve the non-convex optimisation of GLPUAL was proposed also based on ADMM with the non-linear decision boundary generated in the original feature space for satisfactory classification on the trifurcated PU dataset. Fourthly, GLPUAL was assessed by the experiments on synthetic datasets and real datasets.

Chapter 4

Elastic GLPUAL (E-GLPUAL) and Elastic Kernel Free GLPUAL (EKF-GLPUAL)

4.1 Introduction

Recall the objective function of GLPUAL from Equation 3.2:

$$\begin{aligned}\arg\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C_p\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]_+ \\ + C_u[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\ + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0).\end{aligned}$$

There is only the L2-norm regularised term for the model parameters in the objective function of GLPUAL. In this case, it is hard for GLPUAL to compress the model parameters of irrelevant features to zero [46]. Therefore, it is hard for GLPUAL to rule irrelevant features out, and the F1-score of GLPUAL will increase on the model containing more irrelevant features. Moreover, [47] noted that the SVM-based methods with kernel trick applied also tend to have relatively worse performance on the datasets containing irrelevant features. Therefore, GLPUAL with kernel trick applied for non-linear decision boundary in the original feature space also suffers from irrelevant features.

Motivated by this issue, in Section 4.2, firstly we introduce the L_1-norm regularised term to the objective function of GLPUAL without kernel trick to construct the elastic net [48] to compress more parameters of irrelevant features to zero, based on the idea of [49]. The proposed method in section 4.2 is termed elastic GLPUAL (E-GLPUAL). The algorithm for the optimisation of E-GLPUAL is also proposed in Section 4.2 based on multi-block ADMM. Then in Section 4.3, experiments were conducted on the synthetic PU datasets with irrelevant features to verify our motivation of E-GLPUAL.

Furthermore, noticing that the kernel trick is not applicable to E-GLPUAL due to the introduced L_1-norm regularised term, which is similar to the issues in [49, 50, 51, 52, 53, 54], we introduced the kernel free techniques from the soft quadratic surface SVM (SQSSVM) [55] to the predictive score function of E-GLPUAL to generate the non-linear decision boundary in Section 4.4. The proposed generalisation of E-GLPUAL is termed as elastic kernel free GLPUAL (EKF-GLPUAL). Besides, we introduced the techniques for the optimisation of group variables [56] to our ADMM-based algorithm of E-GLPUAL to construct the algorithm for the optimisation of EKF-GLPUAL in the rest of Section 4.4. Then we conducted experiments to assess the performance of E-GLPUAL and EKF-GLPUAL on 14 UCI datasets in Section 4.5. Eventually, in Section 4.6, we do theoretical analysis to discuss the gap between the optimal coefficients of several features with these features becoming increasingly similar.

4.2 Methodology and Algorithm of E-GLPUAL for Linear Decision Boundary

The unconstrained optimisation problem of E-GLPUAL is formulated as

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0}&\frac{\lambda_1}{2}\|\boldsymbol{\beta}\|_1+\frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2+C_p\mathbf{1}_p^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}^T\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]_+\\&+C_u[\mathbf{1}_u+(\boldsymbol{X}_{[u]}^T\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]^T[\mathbf{1}_u+(\boldsymbol{X}_{[u]}^T\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]\\&+(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0),\end{aligned}\tag{4.1}$$

with predictive score the same as SVM. i.e.,

$$f(\boldsymbol{x}) = (\boldsymbol{x})^T \boldsymbol{\beta} + \beta_0. \tag{4.2}$$

Though E-GLPUAL can be treated a special case of EKF-GLPUAL as mentioned in Section 4.1, we prefer to preserve the algorithm of E-GLPUAL in this section, since the number of the dual variables needed in this algorithm is much fewer than that in the algorithm of EKF-GLPUAL in Section 4.4 and thus the algorithm of E-GLPUAL converges much faster.

The terms $\|\boldsymbol{\beta}\|_1$ and $\sum_{i=1}^{p} [1 - y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0)]_+$ are not always differentiable in the feasible region of the objective function in Equation 4.1, which bring difficulty to solve the E-GLPUAL by gradient descent. Similar to the case in the optimisation for the objective function of GLPUAL, the following equivalent reformulation can be considered:

$$\begin{aligned} \min_{\boldsymbol{\beta},\beta_0,\boldsymbol{h},\boldsymbol{a},\boldsymbol{t},\boldsymbol{b}} \quad & C_p \mathbf{1}_p^T [\boldsymbol{h}]_+ + C_u \boldsymbol{a}^T \boldsymbol{a} + \boldsymbol{b}^T \boldsymbol{R} \boldsymbol{b} + \frac{\lambda_1}{2} \|\boldsymbol{t}\|_1 + \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2 \\ s.t. \quad & \boldsymbol{h} = \mathbf{1}_p - (\boldsymbol{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0), \\ & \boldsymbol{a} = \mathbf{1}_u + \boldsymbol{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0, \\ & \boldsymbol{t} = \boldsymbol{\beta}, \\ & \boldsymbol{b} = \boldsymbol{X}_{[pu]} \boldsymbol{\beta} + \mathbf{1}_{pu} \beta_0. \end{aligned} \tag{4.3}$$

The constraints of the optimisation in Equation 4.3 can be reformulated into the following form:

$$\begin{bmatrix} \boldsymbol{I}_{n_p} \\ \mathbf{0}_{n_u \times n_p} \\ \mathbf{0}_{m \times n_p} \\ \mathbf{0}_{n_{pu} \times n_p} \end{bmatrix} \boldsymbol{h} + \begin{bmatrix} \mathbf{0}_{n_p \times n_u} \\ \boldsymbol{I}_{n_u} \\ \mathbf{0}_{m \times n_u} \\ \mathbf{0}_{n_{pu} \times n_u} \end{bmatrix} \boldsymbol{a} + \begin{bmatrix} \mathbf{0}_{n_p \times m} \\ \mathbf{0}_{n_u \times m} \\ \boldsymbol{I}_m \\ \mathbf{0}_{n_{pu} \times m} \end{bmatrix} \boldsymbol{t} + \begin{bmatrix} \mathbf{0}_{n_p \times n_{pu}} \\ \mathbf{0}_{n_u \times n_{pu}} \\ \mathbf{0}_{m \times n_{pu}} \\ \boldsymbol{I}_{n_{pu}} \end{bmatrix} \boldsymbol{b} + \begin{bmatrix} \boldsymbol{X}_{[p]}, & \mathbf{1}_p \\ \boldsymbol{X}_{[u]}, & \mathbf{1}_u \\ \boldsymbol{I}_m, & \mathbf{0}_{m \times 1} \\ \boldsymbol{X}_{[pu]}, & 1_{pu} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \mathbf{1}_p \\ \mathbf{1}_u \\ \mathbf{0}_{m \times 1} \\ \mathbf{1}_{pu} \end{bmatrix} \quad , \tag{4.4}$$

where $\mathbf{0}_{k_1 \times k_2}$ denotes a zero matrix of k_1 rows and k_2 columns. The coefficients matrices of the model parameters in Equation 4.4 can be divided into two subsets, i.e.,

$$\left\{\begin{bmatrix}\boldsymbol{I}_{n_p}\\ \mathbf{0}_{n_u\times n_p}\\ \mathbf{0}_{m\times n_p}\\ \mathbf{0}_{n_{pu}\times n_p}\end{bmatrix},\begin{bmatrix}\mathbf{0}_{n_p\times n_u}\\ \boldsymbol{I}_{n_u}\\ \mathbf{0}_{m\times n_u}\\ \mathbf{0}_{n_{pu}\times n_u}\end{bmatrix},\begin{bmatrix}\mathbf{0}_{n_p\times m}\\ \mathbf{0}_{n_u\times m}\\ \boldsymbol{I}_{m}\\ \mathbf{0}_{n_{pu}\times m}\end{bmatrix},\begin{bmatrix}\mathbf{0}_{n_p\times n_{pu}}\\ \mathbf{0}_{n_u\times n_{pu}}\\ \mathbf{0}_{m\times n_{pu}}\\ \boldsymbol{I}_{n_{pu}}\end{bmatrix}\right\} \text{ and } \left\{\begin{bmatrix}\boldsymbol{X}_{[p]}, & \mathbf{1}_p\\ \boldsymbol{X}_{[u]}, & \mathbf{1}_u\\ \boldsymbol{I}_m, & \mathbf{0}_{m\times 1}\\ \boldsymbol{X}_{[pu]}, & 1_{pu}\end{bmatrix}\right\}. \tag{4.5}$$

It should be noted that the objective function of E-GLPUAL in Equation 4.3 is convex for $\boldsymbol{\beta},\beta_0,\boldsymbol{h},\boldsymbol{a},\boldsymbol{t},\boldsymbol{b}$ and the matrices in the first set in Equation 4.5 are orthogonal to each other, which meets the conditions for convergence to apply the direct extension of ADMM of multi-block according to [57].

The Lagrangian function of problem in Equation 4.3 is

$$\begin{aligned}\mathscr{L}(\boldsymbol{\theta}) =& C_p\mathbf{1}_p^T[\boldsymbol{h}]_+ + C_u\boldsymbol{a}^T\boldsymbol{a}+\boldsymbol{b}^T\boldsymbol{R}\boldsymbol{b}+\frac{\lambda_1}{2}\|\boldsymbol{t}\|_1+\frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2\\ &+\boldsymbol{u}_{\boldsymbol{h}}^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)-\boldsymbol{h}]\\ &+\boldsymbol{u}_{\boldsymbol{a}}^T(\mathbf{1}_u+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0-\boldsymbol{a})\\ &+\boldsymbol{v}^T(\boldsymbol{\beta}-\boldsymbol{t})\\ &+\boldsymbol{q}^T(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0-\boldsymbol{b}).\end{aligned} \tag{4.6}$$

where $\boldsymbol{\theta}=\{\boldsymbol{\beta},\beta_0,\boldsymbol{h},\boldsymbol{a},\boldsymbol{t},\boldsymbol{b},\boldsymbol{u}_{\boldsymbol{h}},\boldsymbol{u}_{\boldsymbol{a}},\boldsymbol{v},\boldsymbol{q}\}$, and $\boldsymbol{u}_{\boldsymbol{h}}$, $\boldsymbol{u}_{\boldsymbol{a}}$, $\boldsymbol{v}$ and $\boldsymbol{q}$ are dual variables. Then the augmented Lagrangian function is given as

$$\begin{aligned}\mathscr{L}_a(\boldsymbol{\theta}) =& \mathscr{L}(\boldsymbol{\theta})+\frac{\mu_1}{2}\left\|\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)-\boldsymbol{h}\right\|_2^2\\ &+\frac{\mu_2}{2}\left\|\mathbf{1}_u+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0-\boldsymbol{a}\right\|_2^2\\ &+\frac{\mu_3}{2}\|\boldsymbol{\beta}-\boldsymbol{t}\|_2^2\\ &+\frac{\mu_4}{2}\left\|\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0-\boldsymbol{b}\right\|_2^2.\end{aligned} \tag{4.7}$$

According to the ADMM, the solution of the E-GLPUAL can be found by solving the following updates of iteration until convergence:

$$\begin{aligned}
&(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)}) = \arg\min_{\boldsymbol{\beta},\beta_0} \mathscr{L}_a(\boldsymbol{\beta},\beta_0,\boldsymbol{h}^{(k)},\boldsymbol{a}^{(k)},\boldsymbol{t}^{(k)},\boldsymbol{b}^{(k)},\boldsymbol{u}_{\boldsymbol{h}}^{(k)},\boldsymbol{u}_{\boldsymbol{a}}^{(k)},\boldsymbol{v}^{(k)},\boldsymbol{q}^{(k)}),\\
&\boldsymbol{h}^{(k+1)} = \arg\min_{\boldsymbol{h}} \mathscr{L}_a(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)},\boldsymbol{h},\boldsymbol{a}^{(k)},\boldsymbol{t}^{(k)},\boldsymbol{b}^{(k)},\boldsymbol{u}_{\boldsymbol{h}}^{(k)},\boldsymbol{u}_{\boldsymbol{a}}^{(k)},\boldsymbol{v}^{(k)},\boldsymbol{q}^{(k)}),\\
&\boldsymbol{a}^{(k+1)} = \arg\min_{\boldsymbol{a}} \mathscr{L}_a(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)},\boldsymbol{h}^{(k+1)},\boldsymbol{a},\boldsymbol{t}^{(k)},\boldsymbol{b}^{(k)},\boldsymbol{u}_{\boldsymbol{h}}^{(k)},\boldsymbol{u}_{\boldsymbol{a}}^{(k)},\boldsymbol{v}^{(k)},\boldsymbol{q}^{(k)}),\\
&\boldsymbol{t}^{(k+1)} = \arg\min_{\boldsymbol{t}} \mathscr{L}_a(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)},\boldsymbol{h}^{(k+1)},\boldsymbol{a}^{(k+1)},\boldsymbol{t},\boldsymbol{b}^{(k)},\boldsymbol{u}_{\boldsymbol{h}}^{(k)},\boldsymbol{u}_{\boldsymbol{a}}^{(k)},\boldsymbol{v}^{(k)},\boldsymbol{q}^{(k)}),\\
&\boldsymbol{b}^{(k+1)} = \arg\min_{\boldsymbol{b}} \mathscr{L}_a(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)},\boldsymbol{h}^{(k+1)},\boldsymbol{a}^{(k+1)},\boldsymbol{t}^{(k+1)},\boldsymbol{b},\boldsymbol{u}_{\boldsymbol{h}}^{(k)},\boldsymbol{u}_{\boldsymbol{a}}^{(k)},\boldsymbol{v}^{(k)},\boldsymbol{q}^{(k)}),\\
&\boldsymbol{u}_{\boldsymbol{h}}^{(k+1)} = \boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}],\\
&\boldsymbol{u}_{\boldsymbol{a}}^{(k+1)} = \boldsymbol{u}_{\boldsymbol{a}}^{(k)} + \mu_2(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a}^{(k+1)}),\\
&\boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)} + \mu_3(\boldsymbol{\beta}^{(k+1)} - \boldsymbol{t}^{(k+1)}),\\
&\boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \mu_4(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_{pu}\beta_0^{(k+1)} - \boldsymbol{b}^{(k+1)}).
\end{aligned} \tag{4.8}$$

4.2.1 Update of β and β_0

The update of $\boldsymbol{\beta}$ and β_0 is

$$\begin{aligned}
(\boldsymbol{\beta}^{(k+1)},\beta_0^{(k+1)}) = \arg\min_{\boldsymbol{\beta},\beta_0} &\frac{\lambda_2}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \boldsymbol{u_h}^{(k)^T}[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}]\\
&+ \boldsymbol{u_a}^{(k)^T}(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0 - \boldsymbol{a}^{(k)}) + \boldsymbol{v}^{(k)^T}(\boldsymbol{\beta} - \boldsymbol{t}^{(k)})\\
&+ \boldsymbol{q}^{(k)^T}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{b}^{(k)}) + \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}\right\|_2^2\\
&+ \frac{\mu_2}{2}\left\|\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0 - \boldsymbol{a}^{(k)}\right\|_2^2 + \frac{\mu_3}{2}\left\|\boldsymbol{\beta} - \boldsymbol{t}^{(k)}\right\|_2^2\\
&+ \frac{\mu_4}{2}\left\|\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{b}^{(k)}\right\|_2^2,
\end{aligned} \tag{4.9}$$

which is a quadratic optimisation with every term differentiable.

Defining

$$\begin{aligned}
&\boldsymbol{\beta}^* = \begin{bmatrix}\boldsymbol{\beta}\\ \beta_0\end{bmatrix}, \boldsymbol{X}_{[j]}^* = [\boldsymbol{X}_{[j]}, \mathbf{1}_j], j = p,u,pu, \boldsymbol{I}_{k+1}^{[0]} = \begin{bmatrix}\boldsymbol{I}_k & 0\\ 0 & 0\end{bmatrix},\\
&\boldsymbol{v}^{(k)[0]} = \begin{bmatrix}\boldsymbol{v}^{(k)}\\ 0\end{bmatrix}, \boldsymbol{t}^{(k)[0]} = \begin{bmatrix}\boldsymbol{t}^{(k)}\\ 0\end{bmatrix},
\end{aligned}$$

the solution of problem in Equation 4.9 can be obtained by solving the following linear equation w.r.t. $\boldsymbol{\beta}$ and β_0:

$$\begin{aligned}
&\left[(\mu_3+\lambda_2)\boldsymbol{I}_{m+1}^{[0]}+\mu_1\boldsymbol{X}_{[p]}^{*T}\boldsymbol{X}_{[p]}^{*}+\mu_2\boldsymbol{X}_{[u]}^{*T}\boldsymbol{X}_{[u]}^{*}+\mu_4\boldsymbol{X}_{[pu]}^{*T}\boldsymbol{X}_{[pu]}^{*}\right]\boldsymbol{\beta}^{*}\\
&=\boldsymbol{X}_{[p]}^{*T}\boldsymbol{u_h}^{(k)}-\boldsymbol{X}_{[u]}^{*T}\boldsymbol{u_a}^{(k)}-\boldsymbol{v}^{(k)[0]}-\boldsymbol{X}_{[pu]}^{*T}\boldsymbol{q}^{(k)}+\mu_1\boldsymbol{X}_{[p]}^{*T}(\mathbf{1}_p-\boldsymbol{h}^{(k)})\\
&-\mu_2\boldsymbol{X}_{[u]}^{*T}(\mathbf{1}_u-\boldsymbol{a}^{(k)})+\mu_3\boldsymbol{I}_{m+1}^{[0]}\boldsymbol{t}^{(k)[0]}+\mu_4\boldsymbol{X}_{[pu]}^{*T}\boldsymbol{b}^{(k)}.
\end{aligned} \tag{4.10}$$

4.2.2 Update of h

The update of $\boldsymbol{h}$ is

$$\begin{aligned}
\boldsymbol{h}^{(k+1)}=&\arg\min_{\boldsymbol{h}} C_p\mathbf{1}_p^T[\boldsymbol{h}]_+ +\boldsymbol{u}_{\boldsymbol{h}}^{(k)^T}[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)}+\mathbf{1}_p\beta_0^{(k+1)})-\boldsymbol{h}]\\
&+\frac{\mu_1}{2}\left\|\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)}+\mathbf{1}_p\beta_0^{(k+1)})-\boldsymbol{h}\right\|_2^2,
\end{aligned} \tag{4.11}$$

which is equivalent to solve the problem

$$\min_{\boldsymbol{h}}\sum_{i=1}^{n_p}\left\{\frac{C_p}{\mu_1}[h_i]_+ +\frac{1}{2}[1+\frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1}-(\boldsymbol{x}_i^T\boldsymbol{\beta}^{(k+1)}+\beta_0^{(k+1)})-h_i]^2\right\}. \tag{4.12}$$

The problem in Equation 4.12 is the same as the problem in Equation 3.11. Thus, recall function $s_c(d)=\arg\min_x c[x]_+ +\frac{1}{2}(x-d)^2$ and

$$\arg\min_x c[x]_+ +\frac{1}{2}(x-d)^2=\begin{cases} d-c, d>c,\\ 0, 0\le d\le c,\\ d, d<0.\end{cases} \tag{4.13}$$

In this case the *i*th element of $\boldsymbol{h}^{(k+1)}$ in problem in Equation 4.11 is solved as

$$h_i^{(k+1)} = s_{\frac{C_p}{\mu_1}}\left[1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{x}_i^T\boldsymbol{\beta}^{(k+1)} + \beta_0^{(k+1)})\right]. \tag{4.14}$$

4.2.3 Update of a

The update of $\boldsymbol{a}$ is

$$\begin{aligned}\boldsymbol{a}^{(k+1)} = \arg\min_{\boldsymbol{a}} C_u\boldsymbol{a}^T\boldsymbol{a} + \boldsymbol{u}_{\boldsymbol{a}}^{(k)^T}[\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a}] \\ + \frac{\mu_2}{2}\left\|(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a})\right\|_2^2,\end{aligned} \tag{4.15}$$

which is equivalent to solving

$$\begin{aligned}\min_{\boldsymbol{a}} \frac{\mu_2}{2}(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a})^T(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a}) \\ + C_u\boldsymbol{a}^T\boldsymbol{a} - \boldsymbol{u}_{\boldsymbol{a}}^{(k)^T}\boldsymbol{a}.\end{aligned} \tag{4.16}$$

Problem in Equation 4.16 is also quadratic like problem in Equation 4.9 in Section 4.2.1. Thus we have

$$\boldsymbol{a}^{(k+1)} = \frac{1}{2C_u + \mu_2}[\boldsymbol{u}_{\boldsymbol{a}}^{(k)} + \mu_2(\mathbf{1}_u + \boldsymbol{X}_{[u]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_u\beta_0^{(k+1)})]. \tag{4.17}$$

4.2.4 Update of t

The update of $\boldsymbol{t}$ is

$$\boldsymbol{t}^{(k+1)} = \arg\min_{\boldsymbol{t}} \frac{\lambda_1}{2}\|\boldsymbol{t}\|_1 + \boldsymbol{v}^{(k)^T}(\boldsymbol{\beta}^{(k+1)} - \boldsymbol{t}) + \frac{\mu_3}{2}\left\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{t}\right\|_2^2. \tag{4.18}$$

According to [58], problem in Equation 4.18 can be regarded as the sum of m soft threshold functions, where m denotes the number of features, and hence the ith element of the solution of $\boldsymbol{t}^{(k+1)}$ is

$$t_i^{(k+1)} = \frac{\left|\frac{1}{\mu_3}v_i^{(k)} + \beta_i^{(k+1)}\right|}{\frac{1}{\mu_3}v_i^{(k)} + \beta_i^{(k+1)}} \left[\left|\frac{1}{\mu_3}v_i^{(k)} + \beta_i^{(k+1)}\right| - \frac{\lambda_1}{2\mu_3}\right]_+, \tag{4.19}$$

where $v_i^{(k)}$ is the ith element of $\boldsymbol{v}^{(k)}$ and $\beta_i^{(k+1)}$ is the ith element of $\boldsymbol{\beta}^{(k+1)}$.

4.2.5 Update of b

The update of $\boldsymbol{b}$ is

$$\begin{aligned}\boldsymbol{b}^{(k+1)} = \arg\min_{\boldsymbol{b}} \boldsymbol{b}^T\boldsymbol{R}\boldsymbol{b} + {\boldsymbol{q}^{(k)}}^T(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_{[pu]}\beta_0^{(k+1)} - \boldsymbol{b}) \\ + \frac{\mu_4}{2}\left\|\boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_{[pu]}\beta_0^{(k+1)} - \boldsymbol{b}\right\|_2^2.\end{aligned} \tag{4.20}$$

which is quadratic as the problem in Equation 4.9 and 4.15. Thus $\boldsymbol{b}^{(k+1)}$ can be obtained by the following equation:

$$\boldsymbol{b} = (2\boldsymbol{R} + \mu_4\boldsymbol{I}_{pu})^{-1}\left[\boldsymbol{q}^{(k)} + \mu_4(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_{[pu]}\beta_0^{(k+1)})\right]. \tag{4.21}$$

4.3 Experiments for E-GLPUAL on Synthetic Datasets

In this section, experiments were conducted on the linearly separable synthetic datasets to verify the motivation of E-GLPUAL compared with GLPUAL.

4.3.1 The Generation of Synthetic PN Datasets

The 25 linearly separable synthetic datasets in this section were generated based on the 25 synthetic datasets generated in Section 3.4.1 by the following steps:

1. Standardise the 25 synthetic datasets generated in Section 3.4.1.

2. Add four irrelevant features into the standardised synthetic dataset generated in Step 1. These irrelevant variables follow the i.i.d. standard normal distribution and the quantity of these irrelevant features is twice the number of the features in the standardized synthetic dataset generated in Step 1.

The standardisation in Step 1 is to ensure that the introduced irrelevant features in Step 2 exert sufficient disturbance on the original datasets for model training, compared with the magnitude of these datasets. The same as Section 3.4.1, the 25 synthetic datasets in this section are also divided into five categories, according to the expectation vector $(\mathbf{mean}_{p2}, \mathbf{mean}_{p2})$, $\mathbf{mean}_{p2} = 50, 100, 200, 500, 1000$, of the second positive subset of the original synthetic datasets generated in Section 3.4.1.

4.3.2 Training-Test Split for the Synthetic PU Datasets

The same as Section 3.4.2, both E-GLPUAL and GLPUAL can be applied on the datasets sampled from either single-training-set scenario or case-control scenario as we set the suitable metric for hyper-parameter tuning in practice. In this case, for more intuitive comparison, we split each of the synthetic dataset generated in Section 4.3.1 to construct the PU training and test sets consistent with the single-training-set scenario by the same steps in Section 3.4.2 and obtained 25 pairs of PU training set and test set.

4.3.3 Model Setting

For the hyper-parameter tuning of E-GLPUAL, similar to Section 3.4.3, PUF-score in Equation 3.14 was also utilized as the metric for hyper-parameter tuning. Firstly C_p was fixed to 1 and the number K of the nearest neighbors to 5. Then C_u, σ, λ_1 and λ_2 were determined by 4-fold CV, which reached the highest average PUF-score on the validation sets with the denominator $P[\text{sgn}(f(\boldsymbol{x})) = 1]$ estimated by

$\frac{1}{n_p+n_u}\sum_{\boldsymbol{x}_i \in u \cup p} \daleth(\mathrm{sgn}(f(\boldsymbol{x}_i)) = 1)$ at the single-training-set scenario. More specifically, σ, λ_1 and λ_2 were tuned from the set$\{1,2,3,4,5\} \circ \{0.1,1,10,100\}$. C_u was selected to from the set $\{0.01,0.02,\ldots,0.5\}$ based on the setting in [21]. The hyper-parameters for GLPUAL were tuned by the same strategy in Section 3.4.3.

4.3.4 Results and Analysis

The results of the experiments, on the constructed synthetic PU datasets are summarised in Table 4.1. The results are measured by the average F1-score. Moreover, Table 4.2 summarises the proportion of irrelevant features with coefficients compressed to zero relative to the total number of irrelevant features.

According to the experimental results in Table 4.1, E-GLPUAL outperformed GLPUAL on the all the 5 categories of the synthetic PU datasets with four irrelevant features added. Furthermore the variance of the F1-Score of E-GLPUAL is lower than the variance of the F1-Score of GLPUAL on four out of five categories of the synthetic PU datasets.

Table 4.1: Summary of the average F1-score (%) the standard deviation of the experiments on the synthetic datasets.

mean$_{p2}$	E-GLPUAL	GLPUAL
50	94.33 ± 1.92	87.10 ± 8.47
100	93.74 ± 1.79	90.68 ± 8.47
200	93.54 ± 1.92	84.67 ± 3.06
500	91.41 ± 3.85	79.98 ± 3.50
1000	89.04 ± 2.77	78.15 ± 3.66

According to Table 4.2, nearly all the parameters of the irrelevant features were compressed to zero by E-GLPUAL, while GLPUAL struggled to compress the parameters of irrelevant features to zero. Consequently, E-GLPUAL exhibited higher compression efficiency for irrelevant feature parameters than GLPUAL, thereby enabling E-GLPUAL to outperform GLPUAL on PU datasets with the inclusion of irrelevant features. This verified our motivation to propose E-GLPUAL.

Table 4.2: The average proportion (%) and the standard deviation of the irrelevant features whose coefficients were compressed to zero relative to the total number of irrelevant features.

mean$_{p2}$	E-GLPUAL	GLPUAL
50	0.95 ± 0.11	0.10 ± 0.14
100	0.95 ± 0.11	0.10 ± 0.22
200	1.00 ± 0.00	0.05 ± 0.11
500	1.00 ± 0.00	0.10 ± 0.22
1000	1.00 ± 0.00	0.05 ± 0.11

4.4 Methodology and Algorithm of EKF-GLPUAL for Quadratic Decision Boundary

As discussed in Section 4.1, the introduction of L_1-norm regularised term $||\boldsymbol{\beta}||_1$ makes it impossible to apply kernel trick to E-GLPUAL. As a remedy, kernel free techniques from SQSSVM [55] is introduced to the predictive score function of E-GLPUAL to generate a quadratic decision boundary for the non-linear separable datasets:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \mathrm{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0, \tag{4.22}$$

where matrix $\mathrm{A} = \{\alpha_{ij}\}, i, j = 1, \dots, m$ is an $m \times m$ matrix of the quadratic model parameters to be trained.

We firstly consider the objective function without L_1-norm regularised term of the coefficients:

$$\begin{aligned}
\min_{\mathrm{A},\boldsymbol{\beta},\beta_0} \frac{\lambda_2}{2}\sum_{i=1}^{n_{pu}} ||\mathrm{A}^T\boldsymbol{x}_i + \boldsymbol{\beta}||_2^2 + C_p\sum_{i=1}^{n_p}\left[1 - (\frac{1}{2}\boldsymbol{x}_i^T\mathrm{A}\boldsymbol{x}_i + \boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0)\right]_+ \\
+ C_u\sum_{i=n_p+1}^{n_{pu}}\left[1 + (\frac{1}{2}\boldsymbol{x}_i^T\mathrm{A}\boldsymbol{x}_i + \boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0)\right]^2 + \boldsymbol{b}^T\boldsymbol{R}\boldsymbol{b} \\
s.t. b_i = \frac{1}{2}\boldsymbol{x}_i^T\mathrm{A}\boldsymbol{x}_i + \boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0, i = 1, 2, \dots, n_{pu}.
\end{aligned} \tag{4.23}$$

It should be noted that there exists a way to convert Equation 4.23 to the form similar to Equation 4.1 for the simplification of the optimisation based on the idea in [55] by converting matrix A to a vector. Firstly, consider to place the upper triangle elements of matrix A into $\frac{m^2+m}{2} \times 1$-dimensional vector

$$\boldsymbol{\alpha} = (\alpha_{11}, \alpha_{12}, \ldots \alpha_{1m}, \alpha_{22}, \ldots \alpha_{2m}, \ldots, \alpha_{mm})^T. \tag{4.24}$$

Then, we are going to map the ith feature vector $\boldsymbol{x}_i$ into an $m \times \frac{m^2+m}{2}$ feature matrix $\mathbb{X}_i$ in the following steps, which is illustrated in Figure 4.1:

1. Initialising $\mathbb{X}_i$ as a $\{\mathbf{0}\}_{m \times \frac{m^2+m}{2}}$ matrix.
2. Record the coordinate g of the element in vector $\boldsymbol{\alpha}$ if it is originally from the jth row or column of matrix A. The other coordinate despite j of this element in matrix A is subsequently recorded as k.
3. The gth element in the jth row of matrix $\mathbb{X}_i$ is determined as $\boldsymbol{x}_{ik}$.
4. Repeat Step 2 and Step 3 until the position of all the elements in vector $\boldsymbol{\alpha}$ originally from the jth row or column of matrix A is recorded for fixed j.
5. Repeat Step 4 until j has taken the value from 1 to m.
6. Repeat Step 5 until i has taken the value from 1 to n_{pu}.

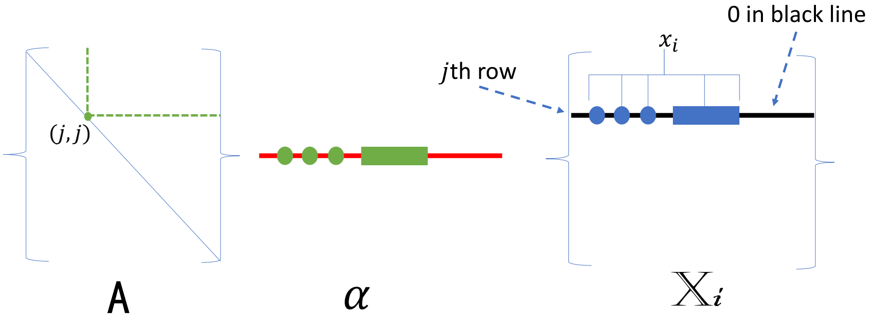

Figure 4.1: Illustration of the steps to construct the jth row of matrix $\mathbb{X}_i$; the elements marked with green circular markers in vector $\boldsymbol{\alpha}$ are equivalent to those represented by the green dashed lines in matrix A; the elements represented by the blue circular markers constitute $\boldsymbol{x}_i$, and their positions in the jth row of matrix $\boldsymbol{X}_i$ correspond to the same positions as the elements represented by the green circular markers in vector $\boldsymbol{\alpha}$.

After obtaining $\boldsymbol{\alpha}$ and $\mathbb{X}_i, i = 1, \ldots, n_{pu}$, define $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T]^T$ and $\mathbb{X}_i^* = [\mathbb{X}_i, \boldsymbol{I}_{m \times m}]$.

Then the optimisation of Equation 4.23 can be converted as

$$\begin{aligned} \min_{\boldsymbol{\alpha}^*, \beta_0} \frac{\lambda_2}{2} \boldsymbol{\alpha}^{*T} \boldsymbol{G} \boldsymbol{\alpha}^* + \boldsymbol{b}^T \boldsymbol{R} \boldsymbol{b} + C_p \mathbf{1}_p^T \left[\mathbf{1}_p - (\boldsymbol{S}_{[p]} \boldsymbol{\alpha}^* + \mathbf{1}_p \beta_0)\right]_+ \\ + C_u \left[\mathbf{1}_u + (\boldsymbol{S}_{[u]} \boldsymbol{\alpha}^* + \mathbf{1}_u \beta_0)\right]^T \left[\mathbf{1}_u + (\boldsymbol{S}_{[u]} \boldsymbol{\alpha}^* + \mathbf{1}_u \beta_0)\right] \\ s.t. \boldsymbol{b} = \boldsymbol{S}_{[pu]} \boldsymbol{\alpha}^* + \mathbf{1}_{pu} \beta_0, \end{aligned} \tag{4.25}$$

where $\boldsymbol{G} = \sum_{i=1}^{n_{pu}} (\mathbb{X}_i^*)^T \mathbb{X}_i^*$, $\boldsymbol{s}_i = \frac{1}{2}[\mathbb{X}_i^{*T} \boldsymbol{x}_i + (\mathbf{0}_{1 \times \frac{m^2+m}{2}}, \boldsymbol{x}_i^T)^T]$, $\boldsymbol{S}_{[p]} = \{\boldsymbol{s}_i^T\}_{n_p \times m^*}$, $\boldsymbol{S}_{[u]} = \{\boldsymbol{s}_i^T\}_{n_u \times m^*}$, $\boldsymbol{S}_{[pu]} = \{\boldsymbol{s}_i^T\}_{n_{pu} \times m^*}$, and $m^* = \frac{m^2+3m}{2}$.

Then we introduce the regularised term of group LASSO [59] to the objective function in Equation 4.25, which is able to generate sparse coefficient vector $\boldsymbol{\alpha}^*$ as the classical L_1-norm with the group-wise penalty realised, i.e., once the parameter of a feature is compressed to 0, all the other parameters of this feature will also be compressed to 0 so that we can abandon this feature from the model [59]. Hence the objective function of EKF-GLPUAL can be obtained as

$$\begin{aligned}\min_{\boldsymbol{\alpha}^*,\beta_0} & \frac{\lambda_2}{2}\boldsymbol{\alpha}^{*T}\boldsymbol{G}\boldsymbol{\alpha}^* + \boldsymbol{b}^T\boldsymbol{R}\boldsymbol{b} + C_p\mathbf{1}_p^T\left[\mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^* + \mathbf{1}_p\beta_0)\right]_+ \\ & + C_u\left[\mathbf{1}_u + (\boldsymbol{S}_{[u]}\boldsymbol{\alpha}^* + \mathbf{1}_u\beta_0)\right]^T\left[\mathbf{1}_u + (\boldsymbol{S}_{[u]}\boldsymbol{\alpha}^* + \mathbf{1}_u\beta_0)\right] \\ & + \frac{\lambda_1}{2}\sum_{i=1}^{m}\|\boldsymbol{\alpha}^*_{[i]}\|_2 \\ & s.t. \boldsymbol{f} = \boldsymbol{S}_{[pu]}\boldsymbol{\alpha}^* + \mathbf{1}_{pu}\beta_0,\end{aligned} \tag{4.26}$$

where the column vector $\boldsymbol{\alpha}^*_{[i]}, i = 1,\dots,m$ is the ith subset of $\boldsymbol{\alpha}^*$ containing the parameters related to the ith feature. The cardinality of $\boldsymbol{\alpha}^*_{[i]}$ for any $i = 1,\dots,m$ is $m+1$. It should be noted that one element in $\boldsymbol{\alpha}^*$ may related to up to 2 features, therefore group overlaps exist among $\boldsymbol{\alpha}^*_{[1]}, \boldsymbol{\alpha}^*_{[2]}, \dots, \boldsymbol{\alpha}^*_{[m]}$, i.e., $\boldsymbol{\alpha}^*_{[i]} \cap \boldsymbol{\alpha}^*_{[j]} \neq \emptyset$, $\forall i, j = 1,\dots,m$. This causes difficulty on ADMM for optimisation so that we need firstly consider the following 5 blocks similar to Equation 4.3:

$$\begin{aligned}\min_{\boldsymbol{\alpha}^*,\beta_0,\boldsymbol{h},\boldsymbol{a},\boldsymbol{t},\boldsymbol{f}} & \frac{\lambda_1}{2}\sum_{i=1}^{m}\|\boldsymbol{t}_{[i]}\|_2 + \frac{\lambda_2}{2}\boldsymbol{\alpha}^{*T}\boldsymbol{G}\boldsymbol{\alpha}^* + \boldsymbol{b}^T\boldsymbol{R}\boldsymbol{b} + C_p\mathbf{1}_p^T[\boldsymbol{h}]_+ + C_u\boldsymbol{a}^T\boldsymbol{a} \\ & s.t. \boldsymbol{h} = \mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^* + \mathbf{1}_p\beta_0), \\ & \boldsymbol{a} = \mathbf{1}_u + (\boldsymbol{S}_{[u]}\boldsymbol{\alpha}^* + \mathbf{1}_u\beta_0), \\ & \boldsymbol{t} = \boldsymbol{D}^*\boldsymbol{\alpha}^*, \\ & \boldsymbol{f} = \boldsymbol{S}_{[pu]}\boldsymbol{\alpha}^* + \mathbf{1}_{pu}\beta_0,\end{aligned} \tag{4.27}$$

where the $m(m+1)$ column vector $\boldsymbol{t} = (\boldsymbol{t}_{[1]}^T, \boldsymbol{t}_{[2]}^T, \dots, \boldsymbol{t}_{[m]}^T)^T$. The cardinality of $\boldsymbol{t}_{[i]}, i = 1,\dots,m$ is $m+1$. Furthermore, $\boldsymbol{D}^*$ is the $m(m+1) \times \frac{m^2+3m}{2}$ design matrix. More specifically, to construct the design matrix $\boldsymbol{D}^*$, firstly we initialise $\boldsymbol{D}^*$ as a $m(m+1) \times \frac{m^2+3m}{2}$ zero matrix. Then if the jth element of $\boldsymbol{\alpha}^*_{[i]}$ is $\boldsymbol{\alpha}^*_k$, the $(j+(i-1)(m+1), k)$ element of $\boldsymbol{D}^*$ will be set to 1; after repeating this Step for all $i = 1,\dots m$ and $j = 1,\dots m+1$, we can finally obtain the design matrix $\boldsymbol{D}^*$ in Equation 4.27. The introduced local variable $\boldsymbol{t}$ can eliminate the group overlaps among the regularised terms of group LASSO, i.e., $\forall i \neq j$, $\boldsymbol{t}_{[i]} \bigcap \boldsymbol{t}_{[j]} = \varnothing$.

Similar to the case in E-GLPUAL for linear decision boundary, the objective function in Equation 4.27 meets the condition to apply ADMM for the optimisation and the 5-block ADMM for EKF-GLPUAL with kernel free techniques introduced can be converted as follows:

$$\begin{aligned}
&(\boldsymbol{\alpha}^{*(k+1)}, \beta_0^{(k+1)}) = \arg\min_{\boldsymbol{\alpha}^*, \beta_0} \mathscr{L}_a(\boldsymbol{\alpha}^*, \beta_0, \boldsymbol{h}^{(k)}, \boldsymbol{a}^{(k)}, \boldsymbol{t}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{u}_{\boldsymbol{h}}^{(k)}, \boldsymbol{u}_{\boldsymbol{a}}^{(k)}, \boldsymbol{v}^{(k)}, \boldsymbol{q}^{(k)}),\\
&\boldsymbol{h}^{(k+1)} = \arg\min_{\boldsymbol{h}} \mathscr{L}_a(\boldsymbol{\alpha}^{*(k+1)}, \beta_0^{(k+1)}, \boldsymbol{h}, \boldsymbol{u}_{\boldsymbol{h}}^{(k)}),\\
&\boldsymbol{a}^{(k+1)} = \arg\min_{\boldsymbol{a}} \mathscr{L}_a(\boldsymbol{\alpha}^{*(k+1)}, \beta_0^{(k+1)}, \boldsymbol{a}, \boldsymbol{u}_{\boldsymbol{a}}^{(k)}),\\
&\boldsymbol{t}^{(k+1)} = \arg\min_{\boldsymbol{t}} \mathscr{L}_a(\boldsymbol{\alpha}^{*(k+1)}, \beta_0^{(k+1)}, \boldsymbol{t}, \boldsymbol{v}^{(k)}),\\
&\boldsymbol{b}^{(k+1)} = \arg\min_{\boldsymbol{b}} \mathscr{L}_a(\boldsymbol{\alpha}^{*(k+1)}, \beta_0^{(k+1)}, \boldsymbol{b}, \boldsymbol{q}^{(k)}),\\
&\boldsymbol{u}_{\boldsymbol{h}}^{(k+1)} = \boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}],\\
&\boldsymbol{u}_{\boldsymbol{a}}^{(k+1)} = \boldsymbol{u}_{\boldsymbol{a}}^{(k)} + \mu_2(\mathbf{1}_u + \boldsymbol{S}_{[u]}\boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_u\beta_0^{(k+1)} - \boldsymbol{a}^{(k+1)}),\\
&\boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)} + \mu_3(\boldsymbol{D}^*\boldsymbol{\alpha}^{*(k+1)} - \boldsymbol{t}^{(k+1)}),\\
&\boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \mu_4(\boldsymbol{S}_{[pu]}\boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_{pu}\beta_0^{(k+1)} - \boldsymbol{b}^{(k+1)}).
\end{aligned} \tag{4.28}$$

4.4.1 Update of α^* and β_0

The update of $\boldsymbol{\alpha}^*$ and β_0 is

$$\begin{aligned}
&\arg\min_{\boldsymbol{\alpha}^*, \beta_0} \frac{\lambda_2}{2}\boldsymbol{\alpha}^{*T}\boldsymbol{G}\boldsymbol{\alpha}^* + \boldsymbol{u_h}^{(k)^T}[\mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^* + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}]\\
&+ \boldsymbol{u_a}^{(k)^T}(\mathbf{1}_u + \boldsymbol{S}_{[u]}\boldsymbol{\alpha}^* + \mathbf{1}_u\beta_0 - \boldsymbol{a}^{(k)}) + \boldsymbol{v}^{(k)^T}(\boldsymbol{\alpha}^* - \boldsymbol{t}^{(k)})\\
&+ \boldsymbol{q}^{(k)^T}(\boldsymbol{S}_{[pu]}\boldsymbol{\alpha}^* + \mathbf{1}_{pu}\beta_0 - \boldsymbol{b}^{(k)}) + \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^* + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k)}\right\|_2^2\\
&+ \frac{\mu_2}{2}\left\|\mathbf{1}_u + \boldsymbol{S}_{[u]}\boldsymbol{\alpha}^* + \mathbf{1}_u\beta_0 - \boldsymbol{a}^{(k)}\right\|_2^2 + \frac{\mu_3}{2}\left\|\boldsymbol{\alpha}^* - \boldsymbol{t}^{(k)}\right\|_2^2\\
&+ \frac{\mu_4}{2}\left\|\boldsymbol{S}_{[pu]}\boldsymbol{\alpha}^* + \mathbf{1}_{pu}\beta_0 - \boldsymbol{b}^{(k)}\right\|_2^2,
\end{aligned} \tag{4.29}$$

which is a quadratic optimisation with every term differentiable.

Thus the model parameters in $\boldsymbol{\alpha}^{**} = \{\boldsymbol{\alpha}^{*T}, \beta_0\}^T$ can be updated by solving

$$\begin{aligned}
&\left[\mu_3 \boldsymbol{D}^{*[0]T}\boldsymbol{D}^{*[0]} + \lambda_2 \boldsymbol{G}^{[0]} + \mu_1 \boldsymbol{S}_{[p]}^{*T}\boldsymbol{S}_{[p]}^{*} + \mu_2 \boldsymbol{S}_{[u]}^{*T}\boldsymbol{S}_{[u]}^{*} + \mu_4 \boldsymbol{S}_{[pu]}^{*T}\boldsymbol{S}_{[pu]}^{*}\right] \boldsymbol{\alpha}^{**} \\
&= \boldsymbol{S}_{[p]}^{*T}\boldsymbol{u_h}^{(k)} - \boldsymbol{S}_{[u]}^{*T}\boldsymbol{u_a}^{(k)} - \boldsymbol{D}^{*[0]T}\boldsymbol{v}^{(k)[0]} - \boldsymbol{S}_{[pu]}^{*T}\boldsymbol{q}^{(k)} + \mu_1 \boldsymbol{S}_{[p]}^{*T}(\mathbf{1}_p - \boldsymbol{h}^{(k)}) \\
&- \mu_2 \boldsymbol{S}_{[u]}^{*T}(\mathbf{1}_u - \boldsymbol{a}^{(k)}) + \mu_3 \boldsymbol{D}^{*[0]T}\boldsymbol{t}^{(k)[0]} + \mu_4 \boldsymbol{S}_{[pu]}^{*T}\boldsymbol{b}^{(k)},
\end{aligned} \tag{4.30}$$

where

$$\boldsymbol{S}_{[j]}^{*} = [\boldsymbol{S}_{[j]}, \mathbf{1}_j], j = p, u, \boldsymbol{D}^{*[0]} = \begin{bmatrix} \boldsymbol{D}^{*} & 0 \\ 0 & 0 \end{bmatrix}.$$

4.4.2 Update of h

The update of $\boldsymbol{h}$ is

$$\begin{aligned}
\boldsymbol{h}^{(k+1)} =& \arg\min_{\boldsymbol{h}} C_p \mathbf{1}_p^T [\boldsymbol{h}]_+ + \boldsymbol{u}_{\boldsymbol{h}}^{(k)^T} [\mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \boldsymbol{h}] \\
&+ \frac{\mu_1}{2} \left\| \mathbf{1}_p - (\boldsymbol{S}_{[p]}\boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \boldsymbol{h} \right\|_2^2,
\end{aligned} \tag{4.31}$$

As mentioned in the optimisation of E-GLPUAL for linear decision boundary, considering the following function:

$$s_c(d) = \arg\min_x c[x]_+ + \frac{1}{2}\|x - d\|_2^2 = \begin{cases} d - c, d > c, \\ 0, 0 \le d \le c, \\ d, d < 0. \end{cases} \tag{4.32}$$

we can solve the ith element of $\boldsymbol{h}^{(k+1)}$ as

$$h_i^{(k+1)} = s_{\frac{C_p}{\mu_1}} \left[1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{s}_i^T \boldsymbol{\alpha}^{*(k+1)} + \beta_0^{(k+1)}) \right]. \tag{4.33}$$

4.4.3 Update of a

The update of $\boldsymbol{a}$ is

$$\begin{aligned}\boldsymbol{a}^{(k+1)} = \arg\min_{\boldsymbol{a}} C_u \boldsymbol{a}^T \boldsymbol{a} + {\boldsymbol{u}_{\boldsymbol{a}}^{(k)}}^T [\mathbf{1}_u + \boldsymbol{S}_{[u]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_u \beta_0^{(k+1)} - \boldsymbol{a}] \\ + \frac{\mu_2}{2} \left\| (\mathbf{1}_u + \boldsymbol{S}_{[u]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_u \beta_0^{(k+1)} - \boldsymbol{a}) \right\|_2^2,\end{aligned} \tag{4.34}$$

This is also a quadratic problem as the update of $\boldsymbol{\beta}$ and β_0. Thus we can find the following solution:

$$\boldsymbol{a}^{(k+1)} = \frac{1}{2C_u + \mu_2} [\boldsymbol{u}_{\boldsymbol{a}}^{(k)} + \mu_2 (\mathbf{1}_u + \boldsymbol{S}_{[u]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_u \beta_0^{(k+1)})]. \tag{4.35}$$

4.4.4 Update of b

The update of $\boldsymbol{b}$ is

$$\begin{aligned}\boldsymbol{b}^{(k+1)} = \arg\min_{\boldsymbol{b}} \boldsymbol{b}^T \boldsymbol{R} \boldsymbol{b} + {\boldsymbol{q}^{(k)}}^T (\boldsymbol{S}_{[pu]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_{[pu]} \beta_0^{(k+1)} - \boldsymbol{b}) \\ + \frac{\mu_4}{2} \left\| \boldsymbol{S}_{[pu]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_{[pu]} \beta_0^{(k+1)} - \boldsymbol{b} \right\|_2^2,\end{aligned} \tag{4.36}$$

which is also a quadratic problem. Thus $\boldsymbol{b}^{(k+1)}$ can be obtained by the following equation:

$$\boldsymbol{b} = (2\boldsymbol{R} + \mu_4 \boldsymbol{I}_{pu})^{-1} \left[\boldsymbol{q}^{(k)} + \mu_4 (\boldsymbol{S}_{[pu]} \boldsymbol{\alpha}^{*(k+1)} + \mathbf{1}_{[pu]} \beta_0^{(k+1)}) \right]. \tag{4.37}$$

4.4.5 Update of *t*

The update of $\boldsymbol{t}$ is

$$\boldsymbol{t}^{(k+1)} = \arg\min_{\boldsymbol{t}} \frac{\lambda_1}{2} \sum_{i=1}^{m} ||\boldsymbol{t}_{[i]}||_2 + \boldsymbol{v}^{(k)^T} (\boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} - \boldsymbol{t}) + \frac{\mu_3}{2} \left\| \boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} - \boldsymbol{t} \right\|_2^2 . \tag{4.38}$$

It is possible to find the closed form of the updated $\boldsymbol{t}$ in each iteration of ADMM since the objective function in Equation 4.38 can be treated as a special scenario of group LASSO regression.

The sub-optimisation in Equation 4.38 is equivalent to

$$\min_{\boldsymbol{t}} \frac{\lambda_1}{2} \sum_{i=1}^{m} ||\boldsymbol{t}_{[i]}||_2 + \frac{\mu_3}{2} \left\| \boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} + \frac{\boldsymbol{v}^{(k)}}{\mu_3} - \boldsymbol{t} \right\|_2^2 . \tag{4.39}$$

This can be regarded as a special scenario of the following group LASSO regression:

$$\frac{\lambda_1}{2\mu_3} \sum_{i=1}^{m} ||\boldsymbol{t}_{[i]}||_2 + \frac{1}{2} ||\boldsymbol{y}_* - \boldsymbol{X}_* \boldsymbol{t}||_2^2 , \tag{4.40}$$

when $\boldsymbol{y}_* = \boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} + \frac{\boldsymbol{v}^{(k)}}{\mu_3}$ and $\boldsymbol{X}_* = \boldsymbol{I}_{m(m+1)}$.

Karush–Kuhn–Tucker (KKT) conditions were proposed as necessary and sufficient conditions for the optimal solution of convex optimisation problems. Based on the techniques for the optimisation of group variables in [56], we can firstly find the KKT conditions for $\boldsymbol{t}_{[i]}$ in Equation 4.40:

$$\begin{aligned} -\boldsymbol{X}_{[i]}^{*T} (\boldsymbol{y}_* - \boldsymbol{X}_* \boldsymbol{t}) + \frac{\lambda_1 \boldsymbol{t}_{[i]}}{2\mu_3 \left\|\boldsymbol{t}_{[i]}\right\|_2} = \boldsymbol{0} \quad & \text{if } \boldsymbol{t}_{[i]} \neq \boldsymbol{0}, \\ \left\| -\boldsymbol{X}_{[i]}^{*T} (\boldsymbol{y}_* - \boldsymbol{X}_* \boldsymbol{t}) \right\|_2 \leqslant \frac{\lambda_1}{2\mu_3} \quad & \text{if } \boldsymbol{t}_{[i]} = \boldsymbol{0}, \end{aligned} \tag{4.41}$$

where $X_{[i]}^* = \boldsymbol{X}_*[\cdot, (i-1)(m+1)+1 : i(m+1)]$. When $\boldsymbol{X}_* = \boldsymbol{I}_{m(m+1)}$ for the sub-optimisation, it is easy to find $\boldsymbol{X}_{[i]}^{*T} \boldsymbol{X}_{[i]}^* = \boldsymbol{I}_{m+1}$.

Therefore, denoting $\mathbb{S}_i = \boldsymbol{X}^{*T}_{[i]} \left(\boldsymbol{y}_* - \boldsymbol{X}^*_{[-i]} \boldsymbol{t}_{[-i]} \right) = \boldsymbol{X}^{*T}_{[i]} \boldsymbol{y}_*$, according to Equation 4.41, we can find

$$\boldsymbol{t}_{[i]} = \left[1 - \frac{\lambda_1}{2\mu_3 \left\| \mathbb{S}_i \right\|_2} \right]_+ \mathbb{S}_i. \tag{4.42}$$

Thus the closed form of the updated $\boldsymbol{t}_{[i]}$ is

$$\boldsymbol{t}^{(k+1)}_{[i]} = \left[1 - \frac{\lambda_1}{2\mu_3 \left\| \boldsymbol{X}^{*T}_{[i]} (\boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} + \frac{\boldsymbol{v}^{(k)}}{\mu_3}) \right\|_2} \right]_+ \boldsymbol{X}^{*T}_{[i]} (\boldsymbol{D}^* \boldsymbol{\alpha}^{*(k+1)} + \frac{\boldsymbol{v}^{(k)}}{\mu_3}). \tag{4.43}$$

4.5 Experiments for EKF-GLPUAL on Real Datasets

In this section experiments on real datasets with irrelevant features added were conducted to compare our proposed EKF-GLPUAL with GLPUAL, and other conventional methods, i.e., uPU and nnPU. More specifically, RBF kernel was applied to GLPUAL for the non-linear decision in the original feature space.

4.5.1 The Source of Datasets

The Experiments were conducted on the 14 UCI datasets used in Section 3.6, i.e., **OD**, **Acc**, **Ecoli**, Pen-Based Recognition of Handwritten Digits (**Pen**), **OR1**, **OR2**, **wifi**, **UMD**,**RD**, **SSMCR**, **Seeds**, **ENB**, **HD**, and **LD**. Due to the computationally singular errors encountered in the current code of EKF-GLPUAL on datasets **PB** and **Avila**, these two datasets were excluded. The details of these 14 datasets are summarised in Table 4.3.

4.5.2 The Preprocessing of the Datasets

To make the experiments consistent to the verification of the motivation of EKF-GLPUAL, the 14 datasets were also preprocessed by the following two steps similar to Section 4.3.1:

1. Standardise the 14 real datasets in Table 4.3.

Table 4.3: Summary of the datasets for the verification of the motivation of EKF-GLPUAL.

Dataset	positive instances	negative instances	features
Acc	100 red	100 blue	4
Ecoli	116 im & 52 pp	143 cp & 25 om	6
Pen	200 one & 200 eight	400 four versicolor	16
OR1	301 UK	301 Germany	4
OR2	500 UK	500 Germany	4
SSMCR	391 alive	109 dead	3
OD	100 occupied	300 not occupied	5
UMD	83 Low	63 high	5
Seeds	70 Kama	70 Rosa	7
ENB	144 TypeII	144 Type III	7
wifi	100 Location 2& 100 Location 4	499 Location 1 & 100 Location 3	7
RD	450 Kecimen	450 Besni	7
LD	144 class 1	200 class 2	6
HD	150 absence	119 presence	13

2. Add several irrelevant features into the standardised synthetic dataset generated in Step 1. These irrelevant variables follow the i.i.d. standard normal distribution and the quantity of these irrelevant features is twice the number of the features in the standardized synthetic dataset generated in Step 1.

For the same aim as Section 4.3.1, the standardisation in Step 1 is to ensure that the introduced irrelevant features in Step 2 exert sufficient disturbance on the original datasets for model training, compared with the magnitude of these datasets.

4.5.3 Training-Test Split for the Real PU Datasets

The training-test split for the 14 real datasets preprocessed by the steps in Section 4.5.2 is the same as the training-test split in Section 3.6.2. In this case, there obtained 10 pairs of PU training and test sets for each of the 14 real datasets with a certain label frequency $\gamma = 0.5, 0.25$ at the case-control scenario.

4.5.4 Compared Methods and Model Setting

GLPUAL, uPU and nnPU were also trained on the 14 real datasets as the compared methods with EKF-GLPUAL. GLPUAL serves as the baseline of EKF-GLPUAL.

By fixing C_p to 1 and the number K of the nearest neighbors to 5, C_u, λ_1, λ_2, and σ in the objective functions of EKF-GLPUAL were firstly tuned by 4-

fold CV, which reached the highest average PUF-score in Equation 3.14 with the denominator $P[\text{sgn}(f(\boldsymbol{x})) = 1]$ estimated by $\frac{1}{n_u}\sum_{\boldsymbol{x}_i \in u} \daleth(\text{sgn}(f(\boldsymbol{x}_i)) = 1)$ at the case-control scenario. More specifically, λ_1, λ_2, and σ were tuned from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$ and C_u was selected to from the set $\{0.5, 0.3, 0.1, 0.05, 0.01\}$ based on the setting in [21]. Then λ, σ and C_u were continually tuned following the greedy algorithm based on the average PUF-score on the validation sets as in Section 3.6.4.

The hyper-parameter tuning for GLPUAL, uPU and nnPU is the same as Section 3.6.4.

4.5.5 Results and Analysis

The results of the experiments are summarised in Table 4.4 by average F1-score. The details of the features abandoned by EKF-GLPUAL in each dataset are summarised in Table 4.5.

Firstly, as for the average F1-score in Table 4.4, EKF-GLPUAL performed better than GLPUAL on 18 cases out of totally 28 cases, which indicates that EKF-GLPUAL can do better classification than GLPUAL when there are irrelevant variables in data. More specifically, EKF-GLPUAL worked better not only on the most cases of the 4 trifurcated PU datasets **wifi**, **OR1**, **OR2** and **Pen**, but also on the most cases of the 6 non-trifurcated PU datasets **SSMCR**, **UMD**, **OD**, **ENB**, **seeds**, and **LD**. Secondly, there are in total 19 cases where EKF-GLPUAL perform better than uPU and nnPU. Finally, there are 11 cases where EKF-GLPUAL is the optimal choice among the four methods in the experiments.

The kernel trick made it impossible to observe the parameters of each feature trained by GLPUAL, so that we cannot recognise which feature was abandoned by GLPUAL. In this case, Table 4.5 only summarised the percentage proportion features abandoned by EKF-GLPUAL in each dataset.

There are mainly two issues reflected from Table 4.5. Firstly, though generally EKF-GLPUAL can abandoned several irrelevant features on most of the training sets, on **HD** and **RD**, the proportion of the irrelevant features abandoned by EKF-GLPUAL is lower than 25% in average. This is not a satisfactory result. Considering that EKF-GLPUAL works not as good as GLPUAL on these two datasets, especially on **RD**, one potential reason for the unsatisfactory performance of EKF-GLPUAL on **HD** and **RD** is that the quadratic boundary generated by the kernel-free setting in Equation 4.26 is not suitable for the structure of **HD** and **RD**. Secondly, another issue reflected by Table 4.5 is that even on the datasets with good performance of EKF-GLPUAL, EKF-GLPUAL cannot abandon all the irrelevant features thoroughly. This is not as good as the case of E-GLPUAL on the synthetic datasets in Section 4.3. Therefore, finding the way to make EKF-GLPUAL to abandon all the irrelevant features accurately is a main future work on EKF-GLPUAL.

Table 4.4: The average F1-scores (%) of the classifiers and their standard deviations; for each of the 14 original datasets, the average F1-score (%) and standard deviation in the two rows were obtained under label frequency $\gamma = 0.5, 0.25$, respectively; the results highlighted in blue for EKF-GLPUAL indicate that it is the best among the four methods; the results highlighted in red for EKF-GLPUAL indicate that it outperforms GLPUAL but falls short of uPU and nnPU.

Dataset	EKF-GLPUAL	GLPUAL	uPU	nnPU
OD	100.00 ± 0.00	85.67 ± 5.97	80.00 ± 42.16	100.00 ± 0.00
	91.39 ± 12.26	80.58 ± 7.85	70.00 ± 48.30	100.00 ±0.00
OR1	88.25 ± 3.46	78.40 ± 2.39	15.76 ± 31.60	78.51 ± 6.50
	79.32 ± 12.33	73.69 ± 3.36	19.59 ± 31.28	67.60 ± 6.61
OR2	80.34 ± 8.21	73.36 ± 1.82	71.81 ± 6.23	76.88 ± 4.20
	80.74 ± 7.07	70.17 ± 6.84	69.26 ± 4.52	72.73 ± 3.81
UMD	91.25 ± 3.14	74.39 ± 4.14	100.00 ± 0.00	100.00 ± 0.00
	89.97 ± 4.61	72.44 ± 4.62	100.00 ± 0.00	100.00 ± 0.00
Acc	71.18 ± 5.29	69.97 ± 4.90	19.19 ± 26.41	19.66 ± 27.64
	71.59 ± 5.42	69.81 ± 5.14	21.30 ± 28.73	22.62 ± 30.29
Ecoli	89.01 ± 0.90	91.58 ± 3.62	76.12 ± 5.59	77.66 ± 6.06
	85.40 ± 7.18	83.41 ± 5.70	76.13 ± 6.39	77.73 ± 5.93
ENB	66.63 ± 5.33	45.25 ± 7.35	28.85 ± 20.18	29.33 ± 21.66
	64.44 ± 6.53	44.58 ± 9.35	24.47 ± 28.50	24.88 ± 29.07
HD	75.33 ± 4.29	80.51 ± 3.08	67.50 ± 4.11	70.63 ± 2.07
	76.31 ± 2.95	81.19 ± 5.13	66.69 ± 4.04	68.94 ± 2.80
Pen	88.34 ± 10.83	82.47 ± 21.16	83.70 ± 9.43	84.51 ± 12.85
	85.77 ± 12.54	80.82 ± 23.88	79.82 ± 10.66	80.49 ± 12.31
LD	58.22 ± 5.00	47.34 ± 6.34	10.89 ± 23.54	29.07 ± 25.57
	58.81 ± 4.13	45.43 ± 5.30	9.53 ± 21.69	18.80 ± 24.58
SSMCR	88.21 ± 1.28	82.84 ± 2.75	79.54 ± 1.49	80.72 ± 1.34

Table 4.4 – continued from previous page

Dataset	EKF-GLPUAL	GLPUAL	uPU	nnPU
	87.86 ± 1.91	82.33 ± 2.18	79.59 ± 1.88	81.92 ± 1.41
Seeds	86.25 ± 4.18 89.90 ± 3.43	74.24 ± 7.23 70.78 ± 21.21	84.14 ± 4.02 80.96 ± 5.89	91.38 ± 5.44 85.64 ± 5.23
wifi	66.22 ± 1.40 77.87 ± 2.27	67.09 ± 3.35 63.58 ± 1.51	82.30 ± 2.71 80.26 ± 3.47	89.07 ± 4.59 85.75 ± 6.25
RD	32.13 ± 2.16 31.82 ± 2.58	77.09 ± 2.90 71.83 ± 4.29	68.57 ± 14.65 67.01 ± 14.08	69.94 ± 16.27 69.53 ± 12.73

Table 4.5: The average percentage proportion of the irrelevant features whose parameters were compressed to zero relative to the total number of irrelevant features with the standard deviation.

OD	84.00 ± 6.99 86.00 ± 6.99	SSMCR	55.00 ± 15.81 53.33 ± 20.49
OR1	75.00 ± 11.79 73.75 ± 10.94	UMD	86.00 ± 9.66 76.00 ± 23.66
OR2	52.50 ± 11.49 46.25 ± 13.76	RD	24.29 ± 9.04 15.71 ± 8.11
Pen	70.94 ± 8.21 67.19 ± 6.95	Seeds	47.14 ± 9.04 51.43 ± 11.07
wifi	75.71 ± 6.02 52.86 ± 10.35	Acc	91.25 ± 9.22 85.00 ± 6.45
Ecoli	77.08 ± 11.85 72.92 ± 12.87	HD	17.69 ± 4.87 15.00 ± 6.14
ENB	48.57 ± 7.38 47.86 ± 9.55	LD	56.67 ± 11.65 55.83 ± 11.15

4.6 Some Theoretical Exploration of on Grouping Effect

The grouping effect of a classifier is an advantage of elastic net, indicating that a classifier assign similar coefficients to similar features [48]; this can preserve the feature which have similar contribution to the classification into the classifier and is hence benefit to the model interpretation. Meanwhile, the classifier with only

L_1-norm regularised term for the model parameters preserves only one feature and assign 0 coefficients to the other features similar to this feature. More specifically, the grouping effect of E-GLPUAL can be summarised into the following theorem:

Theorem 2 *Suppose that $\hat{\beta}_i$ and $\hat{\beta}_j$ are the optimal coefficients of E-GLPUAL for the ith and jth features with. Then there is $\left|(\hat{\beta}_i - \hat{\beta}_j)\right| \to 0$ as $\|\boldsymbol{x}_{[pu]\cdot i} - \boldsymbol{x}_{[pu]\cdot j}\|_2 \to 0$.*

As for the proof of Theorem 2, consider the following equivalent reformulation of unconstrained problem in Equation 4.1:

$$\begin{aligned} \min_{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}_{[p]}} \ & \frac{\lambda_1}{2}\|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 + C_p\mathbf{1}_p^T\boldsymbol{\xi}_{[p]} \\ & + C_u[1+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\beta_0]^T[1+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\beta_0] \\ & + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)' \\ s.t. & \begin{cases} (\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)+\boldsymbol{\xi}_{[p]} \succeq \mathbf{1}_p \\ \qquad \boldsymbol{\xi}_{[p]} \succeq 0. \end{cases} \end{aligned} \tag{4.44}$$

where $\boldsymbol{\xi}_{[p]} = (\xi_1, \xi_2, \ldots, \xi_p)^T \in \mathbb{R}^{p\times 1}$ is a vector of slack variables. The Lagrangian function of problem in Equation 4.44 is given as

$$\begin{aligned} l'(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}_{[p]}) = & \frac{\lambda_1}{2}\|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 + C_p\mathbf{1}_p^T\boldsymbol{\xi}_{[p]} \\ & + C_u[1+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\beta_0]^T[1+\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\beta_0] \\ & + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0) \\ & + \boldsymbol{\upsilon}^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)-\boldsymbol{\xi}_{[p]}]-\boldsymbol{\theta}^T\boldsymbol{\xi}_{[p]} \end{aligned}, \tag{4.45}$$

where $\boldsymbol{\upsilon}, \boldsymbol{\theta} \succeq \mathbf{0}$ are the dual variables of constraint $\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{\xi}_{[p]} \preceq 0$ and $-\boldsymbol{\xi}_{[p]} \preceq 0$, respectively.

Suppose that the solution of the optimisation in Equation 4.44 is $(\hat{\boldsymbol{\beta}}, \hat{\beta}_0, \hat{\boldsymbol{\xi}}_{[p]})$. According to the Karush–Kuhn–Tucker (KKT) conditions [60], for $i \neq j$ with $\hat{\beta}_i\hat{\beta}_j > 0$ we have

$$\begin{aligned}\frac{\partial l'(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}_{[p]})}{\partial \beta_i}\Big|_{(\hat{\boldsymbol{\beta}},\hat{\beta}_0,\hat{\boldsymbol{\xi}}_{[p]})} = {} & \frac{\lambda_1}{2}\operatorname{sgn}(\hat{\beta}_i)+\lambda_2\hat{\beta}_i+2C_u\boldsymbol{x}^T_{[u]\cdot i}(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0) \\ & +2\boldsymbol{x}^T_{[pu]\cdot i}\boldsymbol{R}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)-\boldsymbol{x}^T_{[p]\cdot i}\boldsymbol{\upsilon}=0\end{aligned}, \tag{4.46}$$

$$\begin{aligned}\frac{\partial l'(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}_{[p]})}{\partial \beta_j}\Big|_{(\hat{\boldsymbol{\beta}},\hat{\beta}_0,\hat{\boldsymbol{\xi}}_{[p]})} = {} & \frac{\lambda_1}{2}\operatorname{sgn}(\hat{\beta}_j)+\lambda_2\hat{\beta}_j+2C_u\boldsymbol{x}^T_{[u]\cdot j}(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0) \\ & +2\boldsymbol{x}^T_{[pu]\cdot j}\boldsymbol{R}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)-\boldsymbol{x}^T_{[p]\cdot j}\boldsymbol{\upsilon}=0\end{aligned}, \tag{4.47}$$

and

$$\frac{\partial l'(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}_{[p]})}{\partial \boldsymbol{\xi}_{[p]}}\Big|_{(\hat{\boldsymbol{\beta}},\hat{\beta}_0,\hat{\boldsymbol{\xi}}_{[p]})} = C_p\mathbf{1}_p-\boldsymbol{\upsilon}-\boldsymbol{\theta}=\mathbf{0}. \tag{4.48}$$

Combining Equation 4.48 and $\boldsymbol{\upsilon},\boldsymbol{\theta}\succeq\mathbf{0}$, we have

$$\mathbf{0}\preceq\boldsymbol{\upsilon}\preceq C_p\mathbf{1}_p. \tag{4.49}$$

Assume that $\hat{\beta}_i\hat{\beta}_j>0$ and hence $\operatorname{sgn}(\hat{\beta}_i)=\operatorname{sgn}(\hat{\beta}_j)$. Then subtracting Equation 4.47 from Equation 4.46, the following equation can be obtained:

$$\begin{aligned}&\lambda_2(\hat{\beta}_i-\hat{\beta}_j)+2C_u(\boldsymbol{x}^T_{[u]\cdot i}-\boldsymbol{x}^T_{[u]\cdot j})(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0)\\&+2(\boldsymbol{x}^T_{[pu]\cdot i}-\boldsymbol{x}^T_{[pu]\cdot j})\boldsymbol{R}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)-(\boldsymbol{x}^T_{[p]\cdot i}-\boldsymbol{x}^T_{[p]\cdot j})\boldsymbol{\upsilon}=0.\end{aligned} \tag{4.50}$$

Substituting $(\boldsymbol{\beta},\beta_0)$ in Equation 4.1 by $(\boldsymbol{\beta}=\mathbf{0},\beta_0=0)$ and the solution of the optimisation $(\hat{\boldsymbol{\beta}},\hat{\beta}_0)$, respectively, we have

$$\begin{aligned}&\frac{\lambda_1}{2}\left\|\hat{\boldsymbol{\beta}}\right\|_1+\frac{\lambda_2}{2}\left\|\hat{\boldsymbol{\beta}}\right\|_2^2+C_p\mathbf{1}_p^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\hat{\boldsymbol{\beta}}+\hat{\beta}_0\mathbf{1}_p)]_+ +C_u\left\|\mathbf{1}_u+\boldsymbol{X}_{[u]}\hat{\boldsymbol{\beta}}+\hat{\beta}_0\mathbf{1}_u\right\|_2^2\\&+(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{pu}\hat{\beta}_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{pu}\hat{\beta}_0)\leq n_pC_p+n_uC_u.\end{aligned}\tag{4.51}$$

It should be noted that the square root of symmetric matrix $\boldsymbol{R}$ must exist, denoted as $\boldsymbol{R}^{\frac{1}{2}}$, which satisfies $\boldsymbol{R}=\boldsymbol{R}^{\frac{1}{2}^T}\boldsymbol{R}^{\frac{1}{2}}$. Obviously, every term in the left side of inequality 4.51 is non-negative, so that

$$\left\|\mathbf{1}_u+\boldsymbol{X}_{[u]}\hat{\boldsymbol{\beta}}+\hat{\beta}_0\mathbf{1}_u\right\|_2^2\leq n_p\frac{C_p}{C_u}+n_u\tag{4.52}$$

and

$$\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{pu}\hat{\beta}_0)\right\|_2^2\leq n_pC_p+n_uC_u.\tag{4.53}$$

According to Equation 4.50 and the absolute value inequality, we can obtain the following inequality:

$$\begin{aligned}&\left|(\hat{\beta}_i-\hat{\beta}_j)\right|=\frac{1}{\lambda_2}\Big|2C_u(\boldsymbol{x}_{[u]\cdot i}^T-\boldsymbol{x}_{[u]\cdot j}^T)(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0)\\&+2(\boldsymbol{x}_{[pu]\cdot i}^T-\boldsymbol{x}_{[pu]\cdot j}^T)\boldsymbol{R}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)-(\boldsymbol{x}_{[p]\cdot i}^T-\boldsymbol{x}_{[p]\cdot j}^T)\boldsymbol{\upsilon}\Big|\\&\leq\frac{2C_u}{\lambda_2}\left|(\boldsymbol{x}_{[u]\cdot i}-\boldsymbol{x}_{[u]\cdot j})^T(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0)\right|+\frac{1}{\lambda_2}\left|(\boldsymbol{x}_{[p]\cdot i}-\boldsymbol{x}_{[p]\cdot j})^T\boldsymbol{\upsilon}\right|\\&+\frac{2}{\lambda_2}\left|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})^T\boldsymbol{R}^{\frac{1}{2}^T}\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)\right|.\end{aligned}\tag{4.54}$$

Apply the Cauchy-Schwarz inequality to the right side of Inequality 4.54:

$$\begin{aligned}
&\frac{2C_u}{\lambda_2}\left|(\boldsymbol{x}_{[u]\cdot i}-\boldsymbol{x}_{[u]\cdot j})^T(\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0)\right|+\frac{1}{\lambda_2}\left|(\boldsymbol{x}_{[p]\cdot i}-\boldsymbol{x}_{[p]\cdot j})^T\boldsymbol{\upsilon}\right| \\
&+\frac{2}{\lambda_2}\left|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})^T\boldsymbol{R}^{\frac{1}{2}^T}\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)\right| \\
\le&\frac{2C_u}{\lambda_2}\left\|\boldsymbol{x}_{[u]\cdot i}-\boldsymbol{x}_{[u]\cdot j}\right\|_2\left\|\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0\right\|_2+\frac{1}{\lambda_2}\left\|(\boldsymbol{x}_{[p]\cdot i}-\boldsymbol{x}_{[p]\cdot j})\right\|_2\left\|\boldsymbol{\upsilon}\right\|_2 \\
&+\frac{2}{\lambda_2}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)\right\|_2.
\end{aligned} \tag{4.55}$$

Combining Inequality 4.55, Inequality 4.52, Inequality 4.53 and $\boldsymbol{\upsilon}\preceq C_p\mathbf{1}_p$ in Inequality 4.49, we can obtain

$$\begin{aligned}
&\frac{2C_u}{\lambda_2}\left\|\boldsymbol{x}_{[u]\cdot i}-\boldsymbol{x}_{[u]\cdot j}\right\|_2\left\|\mathbf{1}_u+X_{[u]}\hat{\boldsymbol{\beta}}+\mathbf{1}_u\hat{\beta}_0\right\|_2+\frac{1}{\lambda_2}\left\|(\boldsymbol{x}_{[p]\cdot i}-\boldsymbol{x}_{[p]\cdot j})\right\|_2\left\|\boldsymbol{\upsilon}\right\|_2 \\
&+\frac{2}{\lambda_2}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{X}_{[pu]}\hat{\boldsymbol{\beta}}+\mathbf{1}_{[pu]}\hat{\beta}_0)\right\|_2 \\
\le&\frac{2C_u}{\lambda_2}\sqrt{n_p\frac{C_p}{C_u}+n_u}\left\|(\boldsymbol{x}_{[u]\cdot i}-\boldsymbol{x}_{[u]\cdot j})\right\|_2+\frac{2}{\lambda_2}\sqrt{n_pC_p+n_uC_u}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2 \\
&+\frac{\sqrt{n_p}C_p}{\lambda_2}\left\|(\boldsymbol{x}_{[p]\cdot i}-\boldsymbol{x}_{[p]\cdot j})\right\|_2 \\
\le&\frac{2C_u}{\lambda_2}\sqrt{n_p\frac{C_p}{C_u}+n_u}\left\|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2+\frac{2}{\lambda_2}\sqrt{n_pC_p+n_uC_u}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2 \\
&+\frac{\sqrt{n_p}C_p}{\lambda_2}\left\|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2.
\end{aligned} \tag{4.56}$$

Therefore, combining Inequality 4.54, Inequality 4.55 and Inequality 4.56, we can eventually obtain

$$\begin{aligned}
\left|(\hat{\beta}_i-\hat{\beta}_j)\right|\le&\frac{2C_u}{\lambda_2}\sqrt{n_p\frac{C_p}{C_u}+n_u}\left\|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2+\frac{\sqrt{n_p}C_p}{\lambda_2}\left\|(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2 \\
&+\frac{2}{\lambda_2}\sqrt{n_pC_p+n_uC_u}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{x}_{[pu]\cdot i}-\boldsymbol{x}_{[pu]\cdot j})\right\|_2.
\end{aligned} \tag{4.57}$$

According to inequality 4.57, when $\boldsymbol{x}_{[pu]\cdot i} \to \boldsymbol{x}_{[pu]\cdot j}$, the upper bound of $\left|(\hat{\beta}_i - \hat{\beta}_j)\right|$ is tending to 0, hence the grouping effect holds in EKF-GLPUAL

However, if we follow the same steps to verify the grouping effect on EKF-GLPUAL w.r.t. α_i^* and α_j^*, we will obtain:

$$\left|\frac{\hat{\alpha}_i^*}{\|\hat{\boldsymbol{\alpha}}_{[I]}^*\|_2} - \frac{\hat{\alpha}_j^*}{\|\hat{\boldsymbol{\alpha}}_{[J]}^*\|_2} + (\boldsymbol{g}_{i\cdot} - \boldsymbol{g}_{j\cdot})\hat{\boldsymbol{\alpha}}^*\right| \leq \frac{2C_u}{\lambda_2}\sqrt{n_p\frac{C_p}{C_u} + n_u}\left\|(\boldsymbol{s}_{[pu]\cdot i} - \boldsymbol{s}_{[pu]\cdot j})\right\|_2$$
$$+ \frac{\sqrt{n_p}C_p}{\lambda_2}\left\|(\boldsymbol{s}_{[pu]\cdot i} - \boldsymbol{s}_{[pu]\cdot j})\right\|_2 + \frac{2}{\lambda_2}\sqrt{n_pC_p + n_uC_u}\left\|\boldsymbol{R}^{\frac{1}{2}}(\boldsymbol{s}_{[pu]\cdot i} - \boldsymbol{s}_{[pu]\cdot j})\right\|_2, \tag{4.58}$$

where $\alpha_i^* \in \{\boldsymbol{\alpha}_{[I]}^*\}$, $\alpha_j^* \in \{\boldsymbol{\alpha}_{[J]}^*\}$, $\boldsymbol{s}_{[pu]\cdot i}$ is the *i*th column of $\boldsymbol{S}_{[pu]\cdot i}$, $\boldsymbol{g}_{i\cdot}$ is the *i*th row of $\boldsymbol{G}_{[pu]}$. The left side of Inequality 4.58 is much more complex than the left side of 4.57, where it contains not only the parameters α_i^* and α_j^* but also the other parameters belong to $\{\boldsymbol{\alpha}_{[I]}^*\}$ and $\{\boldsymbol{\alpha}_{[J]}^*\}$. Therefore, the case-dependent studies on the grouping effect of EKF-GLPUAL is also regarded as the future work on EKF-GLPUAL.

4.7 Conclusion

In this Chapter, we proposed E-GLPUAL and EKF-GLPUAL for better classification than GLPUAL on the datasets with irrelevant features. Then the algorithms to solve the optimisation for objective function of E-GLPUAL and EKF-GLPUAL were proposed based on ADMM. EKF-GLPUAL was showed to have better performance than GLPUAL on all the synthetic datasets and several real datasets with irrelevant features added, which supported our motivation. However, currently EKF-GLPUAL cannot abandon all the irrelevant features thoroughly. At the end of this chapter, the grouping effect of E-GLPUAL is proved while the grouping effect of EKF-GLPUAL is a much more complex case to be left as the future work for more case-dependent studies.

Chapter 5

Class-Prior-Based GLPUAL (CPB-GLPUAL)

5.1 Introduction

Recall the objective function of GLPUAL for linear decision boundary:

$$
\begin{aligned}
\min_{\boldsymbol{\beta},\beta_0} & \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C_p\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]_+ \\
& + C_u[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\
& + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0),
\end{aligned}
\tag{5.1}
$$

where $C_p = \frac{1}{n_p}c_p$ and $C_u = \frac{1}{n_u}c_u$.

One weakness of GLPUAL is that there are three hyper-parameters C_u, λ, and σ in similarity matrix $\boldsymbol{R}$ to be tuned by CV with hyper-parameter C_p fixed to 1. An increase in the number of hyper-parameters corresponds to an escalation in the complexity of the identification of optimal hyper-parameter configuration and thus the optimal classification performance.

Motivated by this issue, firstly in Section 5.3, we introduced the setting of uPU [23] to the objective function of GLPUAL to propose a new PU classifier with hyper-parameters C_p and C_u to be determined by class prior π for better classification. The proposed classifier is designated as CPB-GLPUAL. Secondly, in Section 5.4, we proposed an algorithm based on ADMM for the non-convex optimisation of

CPB-GLPUAL to obtain the linear decision boundary in the original feature space. Thirdly, experiments on the synthetic datasets were conducted in Section 5.5 to verify our motivation. Fourthly, in Section 5.6, we introduced the kernel trick to CPB-GLPUAL to obtain the non-linear decision boundary in the original feature space. In Section 5.7, we conducted experiments on the real datasets to assess the performance of CPB-GLPUAL with kernel trick applied.

Furthermore, we found that CPB-GLPUAL can exhibit universal consistency, a good property indicating that the 0-1 risk of CPB-GLPUAL converges in probability to the Bayes risk, which is rarely discussed in PU learning. In Section 5.8, we do theoretical analysis for CPB-GLPUAL to prove its universal consistency to Bayes risk and propose a lower bound for the gap between the 0-1 risk of CPB-GLPUAL and Bayes risk.

5.2 Assumption for CPB-GLPUAL

5.2.1 Case-Control Scenario

The case-control scenario assumption was made on the source of datasets that the labeled-positive and unlabeled instances are collected from two independent datasets, respectively [61]. Furthermore, the unlabeled set is assumed to follow the same distribution to the ground-truth population, i.e., the instances from the unlabeled set are i.i.d. sample of the ground-truth population as shown in the following equation:

$$P[\boldsymbol{X}=\boldsymbol{x}|S=-1]=P[\boldsymbol{X}=\boldsymbol{x}]=\pi P[\boldsymbol{X}=\boldsymbol{x}|Y=1]+(1-\pi)P[\boldsymbol{X}=\boldsymbol{x}|Y=0]. \tag{5.2}$$

5.2.2 Known Class Prior

Consistent with the methods based on uPU, the class prior π is assumed to be known for the model training as the setting of [23]. In the experiments of this chapter, π was estimated by calculating the proportion between the amount of positive instances and the amount of negative instances in the unlabeled set since the datasets were assumed from the case-control scenario. In practice, π can be obtained by either prior knowledge of the data source or the class prior estimation methods, e.g., [62], [63],[64], [65] and [43].

5.3 Methodology of CPB-GLPUAL for Linear Decision Boundary in the Original Feature Space

The general form of the asymmetric loss function for an instance (x, y) in the objective function of GLPUAL can be written as

$$l(f(\boldsymbol{x};\boldsymbol{\beta},\beta_0),s) = \begin{cases} [1-\boldsymbol{x}^T\boldsymbol{\beta}-\beta_0]_+, s=1, \\ (1+\boldsymbol{x}^T\boldsymbol{\beta}+\beta_0)^2, s=-1, \end{cases} \tag{5.3}$$

where s is the labeling indicator.

Then recall the objective function for uPU in Equation 2.8 in Section 2.3.4 is

$$\min_{\boldsymbol{\beta}} \pi\hat{L}_p^1(f)+\hat{L}_u^{-1}(f)-\pi\hat{L}_p^{-1}(f), \tag{5.4}$$

where $\hat{L}_p^1(f) = \frac{1}{n_p}\sum_{\boldsymbol{x}\in p} l(f(\boldsymbol{X};\boldsymbol{\beta}),1)$, $\hat{L}_u^{-1}(f) = \frac{1}{n_u}\sum_{\boldsymbol{x}\in u} l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)$ and $\hat{L}_p^{-1}(f) = \frac{1}{n_p}\sum_{\boldsymbol{x}\in p} l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)$. In this case, $\hat{L}_p^1(f)$, $\hat{L}_u^{-1}(f)$ and $\hat{L}_p^{-1}(f)$ are unbiased estimators of $L_p^1(f) = \mathbb{E}_{\boldsymbol{X}\sim\mathbf{P}}[l(f(\boldsymbol{X};\boldsymbol{\beta}),1)]$, $L_u^{-1}(f) = \mathbb{E}[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)]$ and $L_p^{-1}(f) = \mathbb{E}_{X\sim\mathbf{P}}[l(f(\boldsymbol{X};\boldsymbol{\beta}),-1)]$, respectively. The weights of the loss in the objective function of uPU in Equation 5.4 are determined by class prior π.

Therefore, $\frac{1}{n_p}\mathbf{1}_p^T[\mathbf{1}_p-(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]_+$ in the objective function of GLPUAL in Equation 5.1 can be regarded as $\hat{L}_p^1(f)$ in the objective function of uPU in Equation 5.4, while $\frac{1}{n_u}[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]^T[\mathbf{1}_u+(\boldsymbol{X}_{[u]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]$ can be regarded as $\hat{L}_u^{-1}(f)$ in the objective function of uPU. In this case, we need to add $\frac{1}{n_p}[\mathbf{1}_u+(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_p\beta_0)]^T[\mathbf{1}_p+(\boldsymbol{X}_{[p]}\boldsymbol{\beta}+\mathbf{1}_u\beta_0)]$, equivalent to $\hat{L}_p^{-1}(f)$, into the objective function of GLPUAL to construct an objective function as

$$\begin{aligned}
\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + c_p\frac{1}{n_p}\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]_+ \\
+ c_u\frac{1}{n_u}[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\
+ c_{p2}\frac{1}{n_p}[\mathbf{1}_p + (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]^T[\mathbf{1}_p + (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)] \\
+ (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0).
\end{aligned} \tag{5.5}$$

Let $c_u = c$. According to the objective function of uPU in Equation 5.4, c_p and c_{p2} can be determined as πc and $-\pi c$ to make the weighted average of the loss functions disregarding the coefficient c an unbiased and consistent estimator of the expected loss to classify a new instance. Thus the objective function in Equation 5.5 can be transformed to

$$\begin{aligned}
\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \frac{\pi c}{n_p}\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]_+ \\
+ \frac{c}{n_u}[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)]^T[\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)] \\
- \frac{\pi c}{n_p}[\mathbf{1}_p + (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]^T[\mathbf{1}_p + (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)], \\
+ (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0).
\end{aligned} \tag{5.6}$$

There are only two hyper-parameters, i.e., λ, and σ in similarity matrix $\boldsymbol{R}$, need to be selected in the objective function in Equation 5.6 as the hyper-parameter c is fixed to be 1.

It should be noted that there does not exist a certain $\zeta \in \mathbb{R}$ to make the asymmetric loss of GLPUAL in Equation 5.3 always meet the linear-odd condition proposed in [66], i.e.,

$$l(f,1) - l(f,-1) = [1-f]_+ - (1+f)^2 = \begin{cases} -f^2 - 3f \neq -\zeta f, f < 1; \\ -f^2 - 2f - 1 \neq -\zeta f, f >= 1. \end{cases} \tag{5.7}$$

Not satisfying the odd condition can render the objective function in Equation 5.6 non-convex, leading to significant challenges in optimisation [67]. In order to have an algorithm based on ADMM for solving the non-convex optimisation, we replace the squared loss in Equation 5.7 with the absolute loss $l(f,-1) = |1+f|$, which can also make all unlabeled instances contribute to the construction of decision boundary like the squared loss. The relationship between the absolute loss and the squared loss is illustrated in figure 5.1.

In this way, the objective function of CPB-GLPUAL for linear decision boundary can be represented as

$$\begin{aligned}\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \frac{\pi c}{n_p}\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)]_+ + \frac{c}{n_u}\|\mathbf{1}_u + (\boldsymbol{X}_{[u]}\boldsymbol{\beta} + \mathbf{1}_u\beta_0)\|_1 \\ - \frac{\pi c}{n_p}\|\mathbf{1}_p + (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0)\|_1 + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0).\end{aligned} \tag{5.8}$$

The predictive score function of CPB-GLPUAL for instance $\boldsymbol{x}$ is the same as the predictive score function of GLPUAL, i.e.,

$$f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0.$$

5.4 Algorithm of CPB-GLPUAL for Linear Decision Boundary in the Original Feature Space

5.4.1 ADMM for Non-Convex Optimisation

Despite ADMM being initially proposed for convex optimization in [37], in recent years, studies [68, 69, 70, 71, 72, 73] have also explored the convergence conditions of ADMM for non-convex and non-differentiable objective functions. Thus in this section, we proposed an algorithm based on ADMM for the non-convex optimisation

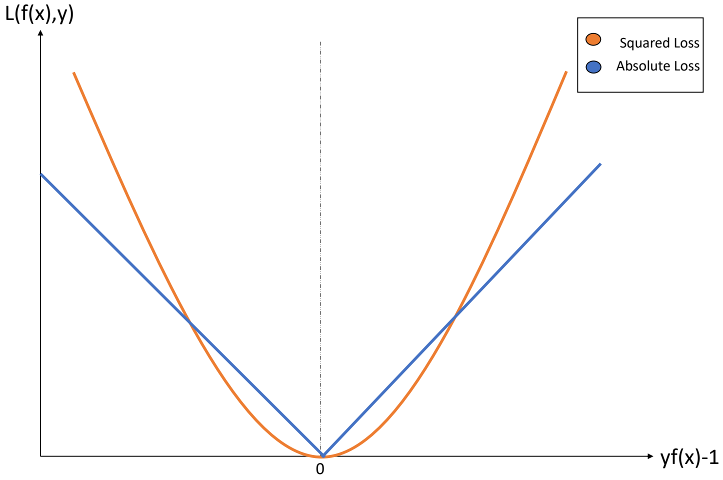

Figure 5.1: The similarity between the absolute loss and the squared loss; x-axis: the distance between the instance and the correct margin boundary; the negative distance indicates that the instance lies on the wrong side of margin while the positive distance indicates that the instance lies on the correct side of the margin; y-axis: the loss of the predictive score function f.

of CPB-GLPUAL for linear decision boundary.

Firstly, let matrix

$$\boldsymbol{C}_n = \begin{bmatrix} -\frac{\pi c}{n_p}\boldsymbol{I}_p & \boldsymbol{0} \\ \boldsymbol{0} & \frac{c}{n_u}\boldsymbol{I}_u, \end{bmatrix}, \tag{5.9}$$

where $\boldsymbol{I}_u$ is an $n_u \times n_u$ identity matrix and $\boldsymbol{I}_p$ is an $n_p \times n_p$ identity matrix, respectively. In this case, the objective function of GLPUAL in Equation 5.8 can be transformed to the following form:

$$\begin{aligned} \min_{\boldsymbol{\beta},\beta_0,\boldsymbol{h},\boldsymbol{a}} & \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0) + \frac{\pi c}{n_p}\boldsymbol{1}_p^T[\boldsymbol{h}]_+ + \boldsymbol{1}_{pu}^T\boldsymbol{C}_n[\boldsymbol{a}]_{++} \\ & s.t.\ \boldsymbol{h} = \boldsymbol{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \boldsymbol{1}_p\beta_0), \\ & \quad \boldsymbol{a} = \boldsymbol{1}_{pu} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0), \end{aligned} \tag{5.10}$$

where $[\boldsymbol{a}]_{++}$ is a column vector and the ith element of $[\boldsymbol{a}]_{++}$ is $|a_i|$

The objective function in Equation 5.10 can be divided into three blocks, i.e., $\frac{\pi c}{n_p}\boldsymbol{1}_p^T[\boldsymbol{h}]_+$, $\boldsymbol{1}_{pu}^T\boldsymbol{C}_n[\boldsymbol{a}]_{++}$ and $\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0)$. $\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \boldsymbol{1}_{pu}\beta_0)$ is convex and Lipschitz differentiable w.r.t. $\boldsymbol{\beta}$ and β_0. $\frac{\pi c}{n_p}\boldsymbol{1}_p^T[\boldsymbol{h}]_+$ is convex but not always differentiable w.r.t.

$\boldsymbol{h}$. $\mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++}$ is neither convex nor always differentiable w.r.t. $\boldsymbol{a}$.

It should be noted that $\mathbf{1}_p^T[\boldsymbol{h}]_+ = \sum_{i=1}^{n_p} \max(0, h_i)$ and $\mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++} = \frac{c}{n_u} \sum_{i=n_p+1}^{n_{pu}} |a_i| - \frac{\pi c}{n_p} \sum_{i=1}^{n_p} |a_i|$; this indicates that $\mathbf{1}_p^T[\boldsymbol{h}]_+$ and $\mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++}$ are piece-wise linear functions for $\boldsymbol{h}$ and $\boldsymbol{a}$, respectively.

Furthermore, $\frac{\pi c \partial \mathbf{1}_p^T[\boldsymbol{h}]_+}{n_p \partial \boldsymbol{h}}$ is a column vector consisting of elements that are either $\frac{\pi c}{n_p}$ or 0. $\frac{\partial \mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++}}{\partial \boldsymbol{a}}$ is a column vector whose elements take value from $\{\frac{\pi c}{n_p}, -\frac{\pi c}{n_p}, \frac{c}{n_u}, -\frac{c}{n_u}\}$. This indicates that $\frac{\pi c \partial \mathbf{1}_p^T[\boldsymbol{h}]_+}{n_p \partial \boldsymbol{h}}$ and $\frac{\partial \mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++}}{\partial \boldsymbol{a}}$ are bounded in any bounded set.

According to [68], for the non-convex objective function which has optimal solution to be solved via ADMM, the non-convex blocks and the blocks not always differentiable are required to be piece-wise linear and their partial derivatives are required to be bounded in any bounded set. [68] also requires the convex blocks to be Lipschitz differentiable. As discussed above in this section, the three blocks $\mathbf{1}_p^T[\boldsymbol{h}]_+$, $\mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++}$ and $\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T \boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)$ all meet their corresponding requirements, respectively. Therefore, based on the proposed structure of ADMM in [68], we propose the following algorithm to solve the optimisation of CPB-GLPUAL in Equation 5.8.

Firstly, the Lagrangian function of the objective function of CPB-GLPUAL in Equation 5.10 is

$$\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}_{cpb}) = & \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T \boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0) \\
& + \frac{\pi c}{n_p}\mathbf{1}_p^T[\boldsymbol{h}]_+ + \mathbf{1}_{pu}^T \boldsymbol{C}_n[\boldsymbol{a}]_{++} \\
& + \boldsymbol{u}_{\boldsymbol{h}}^T[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}] \\
& + \boldsymbol{u}_{\boldsymbol{a}}^T(\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{a}) \\
& s.t.\ \boldsymbol{h} = \mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0), \\
& \quad \boldsymbol{a} = \mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0,
\end{aligned} \tag{5.11}$$

where $\boldsymbol{\theta}_{cpb} = \{\boldsymbol{\beta}, \beta_0, \boldsymbol{h}, \boldsymbol{a}, \boldsymbol{u}_{\boldsymbol{h}}, \boldsymbol{u}_{\boldsymbol{a}}\}$, $\boldsymbol{u}_{\boldsymbol{h}}$ and $\boldsymbol{u}_{\boldsymbol{a}}$ are dual variables. Furthermore, the augmented Lagrangian function of CPB-GLPUAL is defined as

$$\begin{aligned}\mathscr{L}_a(\boldsymbol{\theta}_{cpb}) =& \mathscr{L}(\boldsymbol{\theta}_{cpb}) + \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}\right\|_2^2 \\ &+ \frac{\mu_2}{2}\left\|\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{a}\right\|_2^2.\end{aligned} \tag{5.12}$$

The Lagrangian function and augmented Lagrangian function of CPB-GLPUAL is similar to the ones of GLPUAL. Differently, according to [68], we need to optimise the non-convex blocks at first, the convex blocks but not always differentiable at second and the convex Lipschitz differentiable blocks at last. Hence, the optimisation of the CPB-GLPUAL for linear decision boundary can be handled by iteratively solving the following updates until convergence:

$$\begin{aligned}
\boldsymbol{a}^{(k+1)} &= \arg\min_{\boldsymbol{a}} \mathscr{L}_a(\boldsymbol{\beta}^{(k)}, \beta_0^{(k)}, \boldsymbol{h}^{(k)}, \boldsymbol{a}, \boldsymbol{u}_{\boldsymbol{h}}^{(k)}, \boldsymbol{u}_{\boldsymbol{a}}^{(k)}), \\
\boldsymbol{h}^{(k+1)} &= \arg\min_{\boldsymbol{h}} \mathscr{L}_a(\boldsymbol{\beta}^{(k)}, \beta_0^{(k)}, \boldsymbol{h}, \boldsymbol{a}^{(k+1)}, \boldsymbol{u}_{\boldsymbol{h}}^{(k)}, \boldsymbol{u}_{\boldsymbol{a}}^{(k)}), \\
(\boldsymbol{\beta}^{(k+1)}, \beta_0^{(k+1)}) &= \arg\min_{\boldsymbol{\beta},\beta_0} \mathscr{L}_a(\boldsymbol{\beta}, \beta_0, \boldsymbol{h}^{k+1}, \boldsymbol{a}^{(k+1)}, \boldsymbol{u}_{\boldsymbol{h}}^{(k+1)}, \boldsymbol{u}_{\boldsymbol{a}}^{(k)}), \\
\boldsymbol{u}_{\boldsymbol{h}}^{(k+1)} &= \boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}], \\
\boldsymbol{u}_{\boldsymbol{a}}^{(k+1)} &= \boldsymbol{u}_{\boldsymbol{a}}^{(k)} + \mu_2[\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k+1)} + \mathbf{1}_{pu}\beta_0^{(k+1)} - \boldsymbol{a}^{(k+1)}].
\end{aligned} \tag{5.13}$$

5.4.2 Update of *a*

According to Equation 5.12, the update of a is to solve

$$\begin{aligned}\boldsymbol{a}^{(k+1)} =& \arg\min_{\boldsymbol{a}} \frac{1}{\mu_2}\mathbf{1}_{pu}^T\boldsymbol{C}_n[\boldsymbol{a}]_{++} + \frac{{\boldsymbol{u}_{\boldsymbol{a}}^{(k)}}^T}{\mu_2}(\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k)} + \mathbf{1}_{pu}\beta_0^{(k)} - \boldsymbol{a}) \\ &+ \frac{1}{2}\left\|\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta}^{(k)} + \mathbf{1}_{pu}\beta_0^{(k)} - \boldsymbol{a}\right\|_2^2.\end{aligned} \tag{5.14}$$

This is equivalent to optimise

$$\arg\min_{\boldsymbol{a}} \frac{1}{\mu_2} \mathbf{1}_{pu}^T \boldsymbol{C}_n [\boldsymbol{a}]_{++} + \frac{1}{2} \left\| \mathbf{1}_{pu} + \frac{\boldsymbol{u}_{\boldsymbol{a}}^{(k)}}{\mu_2} + \boldsymbol{X}_{[pu]} \boldsymbol{\beta}^{(k)} + \mathbf{1}_{pu} \beta_0^{(k)} - \boldsymbol{a} \right\|_2^2 . \tag{5.15}$$

Noted that the terms containing $a_i, i = 1, 2., \ldots, n_{pu}$ in Equation 5.15 do not contain other elements of $\boldsymbol{a}$, we can solve the update of $a_1^{(k+1)}, a_2^{(k+1)}, \ldots, a_{n_{pu}}^{(k+1)}$ independently.

For $a_i^{(k+1)}, i = 1, 2., \ldots, n_p$, the objective function is

$$\frac{-\pi c}{\mu_2 n_p} |a_i| + \frac{1}{2} \left(1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)} - a_i \right)^2 . \tag{5.16}$$

To minimise Equation 5.16, we can consider the following function w.r.t. x:

$$j_p |x| + \frac{1}{2} (x - d_p)^2, j_p < 0, \tag{5.17}$$

where j_p and d_p are constants. There are four cases of the threshold function in Equation 5.17, as illustrated in Figure 5.2. Thus we can define

$$g_{j_p}^{[1]}(d_p) = \arg\min_x j_p |x| + \frac{1}{2} (x - d_p)^2 = \begin{cases} d_p + j_p, & d_p < 0, \\ d_p - j_p, & d_p \geq 0. \end{cases} \tag{5.18}$$

Therefore the solution of $a_i^{(k+1)}, i = 1, 2., \ldots, n_p$ can be obtained via computing

$$a_i^{(k+1)} = g_{\frac{-\pi c}{\mu_2 n_p}}^{[1]} (1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)}), i = 1, 2., \ldots, n_p. \tag{5.19}$$

As for the update of $a_i^{(k+1)}, i = n_p + 1, n_p + 2, \ldots, n_{pu}$, we need to separately

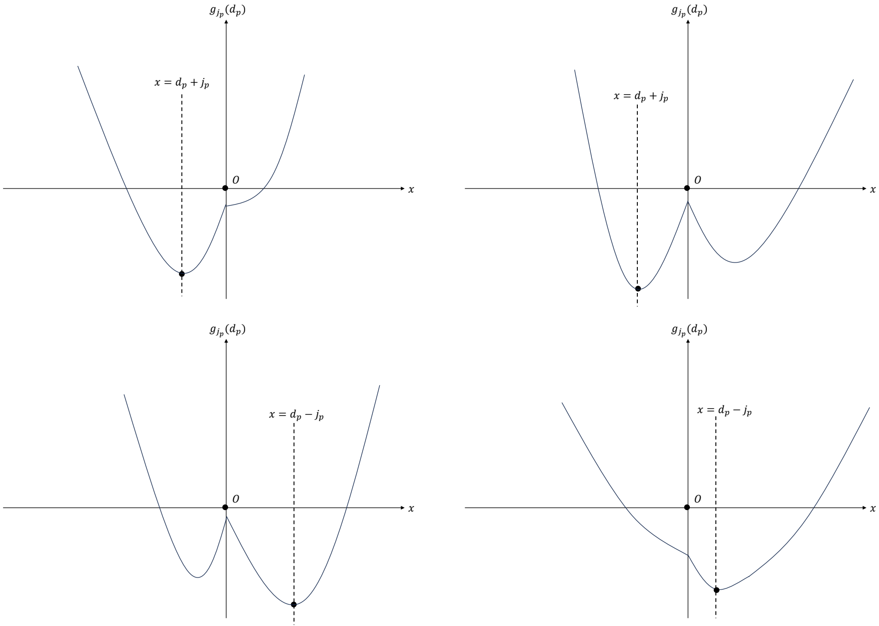

Figure 5.2: The four cases of the threshold function $j_p|x| + \frac{1}{2}(x - d_p)^2, j_p < 0$ w.r.t. x; top left: $d_p < j_p$; top right: $j_p < d_p < 0$; bottom left: $0 < d_p < -j_p$; bottom right: $d_p > -j_p$.

solve

$$\frac{c}{\mu_2 n_u}|a_i| + \frac{1}{2}\left(1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)} - a_i\right)^2. \tag{5.20}$$

To minimise Equation 5.20 we can consider the following function w.r.t. x:

$$j_u|x| + \frac{1}{2}(x - d_u)^2, j_u > 0, \tag{5.21}$$

where j_u and d_u are constants. The three cases of the threshold function in Equation 5.21 are as follows

$$\arg\min_x j_u|x| + \frac{1}{2}(x - d_u)^2 = \begin{cases} d_u + j_u, & d_u < -j_u, \\ 0, & -j_u \le d_u \le j_u, \\ d_u - j_u, & d_u < -j_u. \end{cases} \tag{5.22}$$

Thus, by defining $g_{j_u}^{[2]}(d_u) = \arg\min_x j_u|x| + \frac{1}{2}(x - d_u)^2$, $a_i^{(k+1)}, i = n_p + 1, n_p +$

$2, \ldots, n_{pu}$ can be solved via computing

$$a_i^{(k+1)} = g^{[2]}_{\frac{c}{\mu_2 n_p}}(1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)}), i = n_p + 1, n_p + 2, \ldots, n_p + n_u. \quad (5.23)$$

5.4.3 Update of $\boldsymbol{h}$

The update of $\boldsymbol{h}$ is to solve

$$\begin{aligned} \boldsymbol{h}^{(k+1)} = & \arg\min_{\boldsymbol{h}} \frac{\pi c}{n_p} \mathbf{1}_p^T [\boldsymbol{h}]_+ + \boldsymbol{u}_{\boldsymbol{h}}^{(k)^T} [\mathbf{1}_p - (\boldsymbol{X}_{[p]} \boldsymbol{\beta}^{(k)} + \mathbf{1}_p \beta_0^{(k)}) - \boldsymbol{h}] \\ & + \frac{\mu_1}{2} \left\| \mathbf{1}_p - (\boldsymbol{X}_{[p]} \boldsymbol{\beta}^{(k)} + \mathbf{1}_p \beta_0^{(k)}) - \boldsymbol{h} \right\|_2^2, \end{aligned} \quad (5.24)$$

which is equivalent to solve the problem

$$\min_{\boldsymbol{h}} \sum_{i=1}^{n_p} \left\{ \frac{\pi c}{n_p \mu_1} [h_i]_+ + \frac{1}{2} [1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)}) - h_i]^2 \right\}. \quad (5.25)$$

The way to minimise the threshold function in Equation 5.25 is the same as the the way to solve the problem in Equation 3.11. Recall function $j[x]_+ + \frac{1}{2}(x-d)^2, j > 0$ and

$$s_j(d) = \arg\min_x j[x]_+ + \frac{1}{2}(x-d)^2 = \begin{cases} d - j, d > j, \\ 0, 0 \le d \le j, \\ d, d < 0, \end{cases} \quad (5.26)$$

$h_i, i = 1, 2., \ldots, n_p$ can be updated via computing

$$h_i^{(k+1)} = s_{\frac{\pi c}{n_p}} \left[1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{x}_i^T \boldsymbol{\beta}^{(k)} + \beta_0^{(k)}) \right] \quad (5.27)$$

5.4.4 Update of β and β_0

The update of $\boldsymbol{\beta}$ and β_0 is to solve

$$\begin{aligned}
&\arg\min_{\boldsymbol{\beta},\beta_0} \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + (\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0) \\
&+ \boldsymbol{u_h}^{(k)^T}[\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k+1)}] + \frac{\mu_1}{2}\left\|\mathbf{1}_p - (\boldsymbol{X}_{[p]}\boldsymbol{\beta} + \mathbf{1}_p\beta_0) - \boldsymbol{h}^{(k+1)}\right\|_2^2 \\
&+ \boldsymbol{u_a}^{(k)^T}[\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{a}^{(k+1)}] + \frac{\mu_2}{2}\left\|\mathbf{1}_{pu} + \boldsymbol{X}_{[pu]}\boldsymbol{\beta} + \mathbf{1}_{pu}\beta_0 - \boldsymbol{a}^{(k+1)}\right\|_2^2,
\end{aligned} \tag{5.28}$$

which is a quadratic function as discussed in 3.3.2. Therefore we can solve the optimisation in Equation 5.28 via the KKT condition directly.

Let $\boldsymbol{I}_k, \forall k \in \mathbb{Z}$ denote a $k \times k$ identity matrix. By defining

$$\begin{aligned}
\boldsymbol{M}_{11} &= \lambda\boldsymbol{I}_m + 2\boldsymbol{X}_{[pu]}^T\boldsymbol{R}\boldsymbol{X}_{[pu]} + \mu_1\boldsymbol{X}_{[p]}^T\boldsymbol{X}_{[p]} + \mu_2\boldsymbol{X}_{[pu]}^T\boldsymbol{X}_{[pu]}, \\
\boldsymbol{M}_{12} &= 2\boldsymbol{X}_{[pu]}^T\boldsymbol{R}\mathbf{1}_{pu} + \mu_1\boldsymbol{X}_{[p]}^T\mathbf{1}_p + \mu_2\boldsymbol{X}_{[pu]}^T\mathbf{1}_{pu}, \\
\boldsymbol{M}_{21} &= \boldsymbol{M}_{12}^T, \\
M_{22} &= 2\mathbf{1}_{pu}^T\boldsymbol{R}\mathbf{1}_{pu} + \mu_1 n_p + \mu_2(n_p + n_u), \\
\boldsymbol{m}_1 &= \boldsymbol{X}_{[p]}^T\boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1\boldsymbol{X}_{[p]}^T(\mathbf{1}_p - \boldsymbol{h}^{(k+1)}) - \boldsymbol{X}_{[pu]}^T\boldsymbol{u}_{\boldsymbol{a}}^{(k)} - \mu_2\boldsymbol{X}_{[pu]}^T(\mathbf{1}_{pu} - \boldsymbol{a}^{(k+1)}), \\
m_2 &= \mathbf{1}_p^T\boldsymbol{u}_{\boldsymbol{h}}^{(k)} + \mu_1\mathbf{1}_p^T(\mathbf{1}_p - \boldsymbol{h}^{(k+1)}) - \mathbf{1}_{pu}^T\boldsymbol{u}_{\boldsymbol{a}}^{(k)} - \mu_2\mathbf{1}_{pu}^T(\mathbf{1}_{pu} - \boldsymbol{a}^{(k+1)}),
\end{aligned} \tag{5.29}$$

the solution of problem in Equation 5.28 can be obtained by solving the following linear equation w.r.t. $\boldsymbol{\beta}$ and β_0:

$$\begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}^{(k+1)} \\ \beta_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{m}_1 \\ m_2 \end{bmatrix}. \tag{5.30}$$

5.5 Experiments for CPB-GLPUAL on Synthetic Datasets

In this section, experiments were conducted on the linearly separable synthetic datasets to verify the motivation of CPB-GLPUAL compared with GLPUAL.

5.5.1 The Generation of Synthetic PN Datasets

Considering the baseline GLPUAL is relatively good at the linearly separable dataset following the pattern in Figure 3.2, we use the same way to generate synthetic datasets as in Section 4.3.1. Also the same as Section 3.4.1, the 25 synthetic datasets in this section are also divided into five categories, according to the expectation vector $(\mathbf{mean}_{p2}, \mathbf{mean}_{p2})$, $\mathbf{mean}_{p2} = 50, 100, 200, 500, 1000$, of the second positive subset of the original synthetic datasets generated in Section 3.4.1.

5.5.2 Training-Test Split for the Synthetic PU Datasets

Considering CPB-GLPUAL was proposed at the case-control scenario, we split each of the synthetic dataset generated in Section 4.3.1 to construct the PU training and test sets consistent to the case-control scenario by the same steps as in Section 3.6.2 and obtained 25 pairs of PU training set and test set. More specifically, the value of γ' is set to $\frac{7}{37}$ so that we have the label frequency $\gamma = \gamma'/(0.3\gamma' + 0.7) = 0.25$.

5.5.3 Model Setting

For the hyper-parameter tuning of CPB-GLPUAL, similar to Section 4.3.3, PUF-score in Equation 3.14 was also utilized as the metric for hyper-parameter tuning. Firstly c was fixed to 1 and the number K of the nearest neighbors to 5. Then σ and λ were determined by 4-fold CV, which reached the highest average PUF-score on the validation sets with the denominator $P[\mathrm{sgn}(f(\boldsymbol{x})) = 1]$ estimated by $\frac{1}{n_u}\sum_{\boldsymbol{x}_i \in u} \beth(\mathrm{sgn}(f(\boldsymbol{x}_i)) = 1)$ at the case-control scenario. More specifically, σ and λ were tuned from the set $\{1,2,3,4,5\} \circ \{0.1, 1, 10, 100\}$. The hyper-parameters for GLPUAL were tuned by the same strategy in Section 3.4.3.

5.5.4 Results and Analysis

The results of the experiments, on the constructed synthetic PU datasets are summarised in Table 5.1. The results are measured by the average F1-score. According to the experimental results in Table 5.1, CPB-GLPUAL always has better performance than GLPUAL on the synthetic PU datasets with all the 5 values of $\mathbf{mean}_{p2}$. Therefore it is verified that CPB-GLPUAL can have better performance to generate the linear decision boundary than GLPUAL on the datasets following the pattern in Figure 3.2 with class prior π known.

Table 5.1: Summary of the average F1-score (%) and the standard deviation of the experiments on the synthetic datasets.

$\mathbf{mean}_{p2}$	CPB-GLPUAL	GLPUAL
50	96.41 ± 1.43	93.26 ± 1.80
100	96.67 ± 1.48	93.15 ± 0.98
200	96.03 ± 1.63	94.25 ± 1.60
500	97.02 ± 1.52	92.55 ± 2.07
1000	96.97 ± 1.87	91.51 ± 0.84

5.6 Kernel Trick to CPB-GLPUAL for Non-Linear Decision Boundary

Similar to GLPUAL, the only regularised term in the objective function of CPB-GLPUAL is the L2-norm of $\boldsymbol{\beta}$. Therefore, the kernel trick can also be introduced to CPB-GLPUAL to make CPB-GLPUAL generate non-linear decision boundary for PU classification. The details to achieve this goal is to be discussed in the rest of this section.

Suppose $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{M\times 1}$ be a mapping of the instance vector $\boldsymbol{x}$. Then let $\boldsymbol{\phi}(\boldsymbol{X}_{[k]}) \in \mathbb{R}^{n_k\times r}, k = p, u, pu$ be the mapping of the original data matrix $\boldsymbol{X}_{[k]}$. The ith row of $\boldsymbol{\phi}(\boldsymbol{X}_{[k]})$ is $\phi(\boldsymbol{x}_i)^T \in \mathbb{R}^{1\times r}$. According to Equation 5.29 and Equation 5.30, once we substitute $\boldsymbol{\phi}(\boldsymbol{X}_{pu})$ for $\boldsymbol{X}_{[pu]}$ during the training of classifier, the following necessary condition for the optimal solution of $\boldsymbol{\beta}$ to satisfy can be obtained:

$$\boldsymbol{\beta} = \boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}, \tag{5.31}$$

where

$$\boldsymbol{B} = \boldsymbol{M}_{11} - \frac{\boldsymbol{M}_{12}\boldsymbol{M}_{21}}{M_{22}}, \tag{5.32}$$

and

$$\begin{aligned}\boldsymbol{\Omega} = &\begin{bmatrix} \boldsymbol{u_h} - \mu_1(\mathbf{1}_p - \boldsymbol{h}) - \mu_1\frac{m_2}{M_{22}}\mathbf{1}_p \\ 0 \end{bmatrix} \\ &-[\boldsymbol{u_a} + \mu_2(\mathbf{1}_{pu} - \boldsymbol{a}) + \frac{2m_2}{M_{22}}\boldsymbol{R}\mathbf{1}_{pu} + \frac{\mu_2}{M_{22}}\mathbf{1}_{pu}].\end{aligned} \tag{5.33}$$

Substituting $\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega}$ for $\boldsymbol{\beta}$ in the objective function in Equation 5.8, we have

$$\begin{aligned}&\min_{\boldsymbol{\Omega},\beta_0}\frac{\lambda}{2}\boldsymbol{\Omega}^T\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} \\ &+\frac{\pi c}{n_p}\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{\phi}(\boldsymbol{X}_{[p]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} + \mathbf{1}_p\beta_0)]_+ \\ &+\frac{c}{n_u}\|\mathbf{1}_u + \boldsymbol{\phi}(\boldsymbol{X}_{[u]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} + \beta_0\mathbf{1}_u\|_1 \\ &-\frac{\pi c}{n_p}\|\mathbf{1}_p + \boldsymbol{\phi}(\boldsymbol{X}_{[p]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} + \beta_0\mathbf{1}_p\|_1 \\ &+(\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R} \\ &\quad(\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0),\end{aligned} \tag{5.34}$$

As discussed Section 3.5, according to the two properties for the construction of kernel in Theorem 1, we can obtain

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T =&\boldsymbol{\Phi}'(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})) \\ =&\boldsymbol{\Phi}(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]})\end{aligned} \tag{5.35}$$

and

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T =&\boldsymbol{\Phi}''(\boldsymbol{\phi}(\boldsymbol{X}_{[k]}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})) \\ =&\boldsymbol{\Phi}_2(\boldsymbol{X}_{[k]},\boldsymbol{X}_{[pu]}),\end{aligned} \tag{5.36}$$

where $\boldsymbol{\Phi}'(\boldsymbol{\phi}(\boldsymbol{X}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))$, $\boldsymbol{\Phi}''(\boldsymbol{\phi}(\boldsymbol{X}),\boldsymbol{\phi}(\boldsymbol{X}_{[pu]}))$ are the kernel matrices for $\boldsymbol{\phi}(\boldsymbol{X})$

and $\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})$, and $\boldsymbol{\Phi}(\boldsymbol{X},\boldsymbol{X}_{[pu]}),\boldsymbol{\Phi}_2(\boldsymbol{X},\boldsymbol{X}_{[pu]})$ are the kernel matrices for $\boldsymbol{X}$ and $\boldsymbol{X}_{[pu]}$.

Therefore, the predictive score function in Equation 5.3 for instance $\boldsymbol{x}^*$ of GLPUAL can be transformed to

$$f = \boldsymbol{\Phi}(\boldsymbol{x}^*,\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \beta_0, \tag{5.37}$$

and the objective function of GLPUAL can be eventually transformed to

$$\begin{aligned}
&\min_{\boldsymbol{\Omega},\beta_0} \frac{\lambda}{2}\boldsymbol{\Omega}^T\boldsymbol{\Phi}_2(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \frac{\pi c}{n_p}\mathbf{1}_p^T[\mathbf{1}_p - (\boldsymbol{\Phi}(\boldsymbol{X}_{[p]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_p\beta_0)]_+ \\
&+\frac{c}{n_u}\|\mathbf{1}_u + \boldsymbol{\Phi}(\boldsymbol{X}_{[u]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \beta_0\mathbf{1}_u\|_1 - \frac{\pi c}{n_p}\|\mathbf{1}_p + \boldsymbol{\Phi}(\boldsymbol{X}_{[p]},\boldsymbol{X}_{[p]})\boldsymbol{\Omega} + \beta_0\mathbf{1}_p\|_1 \\
&+(\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]},\boldsymbol{X}_{[pu]})\boldsymbol{\Omega} + \mathbf{1}_{pu}\beta_0),
\end{aligned} \tag{5.38}$$

whose solution is only determined by the kernels.

In this case, the update of $\boldsymbol{a}$ can be written as

$$a_i^{(k+1)} = \begin{cases} g^{[1]}_{\frac{-\pi c}{\mu_2 n_p}}\left[1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{\Phi}(\boldsymbol{x}_i,\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k)} + \beta_0^{(k)}\right], & i = 1,2.,\dots,n_p, \\ g^{[2]}_{\frac{c}{\mu_2 n_p}}\left[1 + \frac{u_{\boldsymbol{a}i}^{(k)}}{\mu_2} + \boldsymbol{\Phi}(\boldsymbol{x}_i,\boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k)} + \beta_0^{(k)}\right], & i = n_p+1, n_p+2,\dots,n_p+n_u. \end{cases} \tag{5.39}$$

Furthermore, the update of $\boldsymbol{h}$ can be reformulated as

$$\boldsymbol{h}_i^{(k+1)} = s_{\frac{\pi c}{n_p}}\left[1 + \frac{u_{\boldsymbol{h}i}^{(k)}}{\mu_1} - (\boldsymbol{\Phi}(\boldsymbol{x}_i,\boldsymbol{X}_{[pu]})\Omega^{(k)} + \beta_0^{(k)})\right], i = 1,2,\dots,n_p. \tag{5.40}$$

Then we can update β_0 via

$$\beta_0^{(k+1)} = \frac{m_2}{M_{22}} - \boldsymbol{Q}_b^{(k+1)}/M_{22}, \tag{5.41}$$

where m_2, M_{22} are not related to $\boldsymbol{X}_{[p]}, \boldsymbol{X}_{[u]}, \boldsymbol{X}_{[pu]}$ and

$$\begin{aligned}\boldsymbol{Q}_b^{(k+1)} =& (2\mathbf{1}_{pu}^T\boldsymbol{R} + \mu_2\mathbf{1}_{pu}^T)\boldsymbol{\Phi}(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} \\ &+ \mu_1\mathbf{1}_p^T\boldsymbol{\Phi}(\boldsymbol{X}_{[p]}, \boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)}.\end{aligned} \tag{5.42}$$

The update of $\boldsymbol{u_h}$ and $\boldsymbol{u_a}$ is

$$\begin{aligned}\boldsymbol{u_h}^{(k+1)} &= \boldsymbol{u_h}^{(k)} + \mu_1[\mathbf{1}_p - (\boldsymbol{\Phi}(\boldsymbol{X}_{[p]}, \boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \boldsymbol{h}^{(k+1)}], \\ \boldsymbol{u_a}^{(k+1)} &= \boldsymbol{u_a}^{(k)} + \mu_2[\mathbf{1}_{pu} + \boldsymbol{\Phi}(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]})\boldsymbol{\Omega}^{(k+1)} + \mathbf{1}_{[pu]]}\beta_0^{(k+1)} - \boldsymbol{a}^{(k+1)}].\end{aligned} \tag{5.43}$$

Thus the update of ADMM for non-linear decision boundary can be summarised into the following steps:

1. Set initial values of $\boldsymbol{\Omega}$, β_0 $\boldsymbol{h}$, $\boldsymbol{u_h}$.

2. Update $\boldsymbol{a}$ via Equation 5.39.

3. Update $\boldsymbol{h}$ via Equation 5.40.

4. Update $\boldsymbol{\Omega}$ and via Equation 5.33 w.r.t. $\boldsymbol{h}^{(k+1)}$, $\boldsymbol{a}^{(k+1)}$, $\boldsymbol{u_h}^{(k)}$ and $\boldsymbol{u_a}^{(k)}$.

5. Update β_0 via Equation 5.41.

6. Update $\boldsymbol{u_h}$ and $\boldsymbol{u_a}$ via Equation 5.43.

7. Repeat Step 2 to Step 6 until convergence.

As discussed in Section 3.5, $\boldsymbol{\Phi}_2(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$ does not directly appear in the update process for the optimisation in this section so that we only need to determine the form of $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$. Moreover, λ either does not appear directly in the above stated update process and it is contained in the matrix $\boldsymbol{B}$ as a part of $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$. Therefore, for convenience, as the case of using the kernel trick in Section 3.4, we use λ to represent the hyper-parameter(s) of the kernel matrix $\boldsymbol{\Phi}(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$.

5.7 Experiments for CPB-GLPUAL on Real Datasets

In this section experiments on real datasets were conducted to compare our proposed CPB-GLPUAL comparing with GLPUAL, and other conventional methods, i.e., uPU and nnPU. More specifically, RBF kernel was applied to GLPUAL for the non-linear decision in the original feature space.

5.7.1 The Source of Datasets and Experimental Design

The Experiments were conducted on the 16 UCI datasets used in Section 3.6, i.e., **OD**, **Acc**, **Ecoli**, Pen-Based Recognition of Handwritten Digits (**Pen**), **OR1**, **OR2**, **wifi**, **UMD**,**RD**, **SSMCR**, **Seeds**, **ENB**, **HD**, **PB**, **Avila**, and **LD**. The details of these 16 datasets are summarised in Table 3.2.

Table 5.2: Summary of the datasets used in experiments for the evaluation of CPB-GLPUAL.

Dataset	positive instances	negative instances	features
Acc	100 red	100 blue	4
Ecoli	116 im & 52 pp	143 cp & 25 om	6
Pen	200 one & 200 eight	400 four versicolor	4
OR1	301 UK	301 Germany	4
OR2	500 UK	500 Germany	4
SSMCR	391 alive	109 dead	3
PB	500 Bull Ring	500 BHMBCCMKT01	3
OD	100 occupied	300 not occupied	5
UMD	83 Low	63 high	5
Seeds	70 Kama	70 Rosa	7
ENB	144 TypeII	144 Type III	7
wifi	100 Location 2& 100 Location 4	499 Location 1 & 100 Location 3	7
Avila	300 E	900 A	10
RD	450 Kecimen	450 Besni	7
LD	144 class 1	200 class 2	6
HD	150 absence	119 presence	13

5.7.2 Training-Test Split for the Real PU Datasets

The training-test split for the 16 real datasets preprocessed by the steps in Section 4.5.2 is the same as the training-test split in Section 3.6.2. In this case, there obtained 10 pairs of PU training and test sets for each of the 16 real datasets with a certain label frequency $\gamma = 0.5, 0.25$ at the case-control scenario.

5.7.3 Compared Methods and Model Setting

GLPUAL, uPU and nnPU were also trained on the 16 real datasets as the compared methods with CPB-GLPUAL. GLPUAL serves as the baseline of CPB-GLPUAL.

By fixing c to 1 and the number K of the nearest neighbors to 5, λ and σ, in the objective functions of EKF-GLPUAL were firstly tuned by 4-fold CV, which reached the highest average PUF-score in Equation 3.14 with the denominator $P[\mathrm{sgn}(f(\boldsymbol{x})) = 1]$ estimated by $\frac{1}{n_u}\sum_{\boldsymbol{x}_i \in u} \beth(\mathrm{sgn}(f(\boldsymbol{x}_i)) = 1)$ at the case-control scenario. More specifically, λ and σ were tuned from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$ based on the setting in [21]. Then λ, σ and C_u were continually tuned following the greedy algorithm based on the average PUF-score on the validation sets as in Section 3.6.4.

The hyper-parameter tuning for GLPUAL, uPU and nnPU is the same as Section 3.6.4.

5.7.4 Results and Analysis

The results of the experiments are summarised in Table 5.3 by average F1-score.

According to the average F1-score in Table 5.3, firstly CPB-GLPUAL performed better than GLPUAL on 17 cases out of 32 cases; this generally supports our motivation that CPB-GLPUAL can outperform GLPUAL when the class prior π is known. More specifically, CPB-GLPUAL achieved better performance than GLPUAL not only on all the cases of the two trifurcated PU datasets **OR1**, **Pen** but also on all the cases of the non-trifurcated PU datasets **ENB**, **HD**, **LD**, **Seeds**, **OD**, and **HD**. This indicates that the structure of the objective function of uPU can improve the performance of the SVM-based PU classifiers once the class prior π is known. Secondly, there are in total 19 cases where CPB-GLPUAL outperformed uPU and nnPU. Finally, there are 14 cases where CPB-GLPUAL is the optimal choice among the four methods in the experiments.

Table 5.3: The average F1-score (%) the standard deviation of the classifiers trained on the 16 real PU datasets; for each of the 16 original datasets, the average F1-score (%) and standard deviation in the two rows were obtained under label frequency $\gamma = 0.5, 0.25$, respectively; the results highlighted in blue for CPB-GLPUAL indicate that it is the best among the four methods; the results highlighted in red for CPB-GLPUAL indicate that it outperforms GLPUAL but falls short of uPU and nnPU.

Dataset	CPB-GLPUAL	GLPUAL	uPU	nnPU
ENB	55.76 ± 9.32 53.92 ± 9.50	42.82 ± 4.76 45.82 ± 7.50	29.58 ± 22.14 26.12 ± 30.53	30.20 ± 23.67 26.88 ± 31.28
HD	88.08 ± 2.50 87.84 ± 2.62	82.72 ± 2.35 81.92 ± 4.03	71.38 ± 4.23 71.01 ± 3.97	74.38 ± 2.19 75.06 ± 2.40
Pen	98.86 ± 1.21 98.20 ± 1.88	92.47 ± 8.13 91.73 ± 9.04	77.76 ± 31.00 72.55 ± 31.03	87.50 ± 14.94 84.06 ± 16.85
LD	53.41 ± 4.54 56.35 ± 4.43	44.24 ± 5.72 36.85 ± 9.97	11.88 ± 25.75 10.15 ± 22.39	31.54 ± 27.79 20.09 ± 26.27
OR1	93.01 ± 3.23 87.38 ± 2.46	90.05 ± 2.25 83.88 ± 5.78	16.64 ± 33.33 20.90 ± 33.12	84.08 ± 6.88 72.06 ± 6.99
Seeds	94.58 ± 1.87 94.28 ± 1.94	92.31 ± 4.86 89.05 ± 5.53	92.37 ± 1.51 86.85 ± 3.12	97.25 ± 3.65 93.08 ± 3.89
wifi	68.90 ± 1.55 68.21 ± 1.29	95.10 ± 1.96 96.69 ± 1.83	91.16 ± 4.29 87.69 ± 2.88	92.17 ± 3.17 89.27 ± 2.61
Avila	0.00 ± 0.00 0.00 ± 0.00	55.82 ± 2.90 50.05 ± 4.22	62.74 ± 8.82 61.30 ± 9.23	63.75 ±9.07 61.00 ± 9.04
OD	100.00 ± 0.00 100.00 ± 0.00	89.00 ± 8.44 95.69 ± 6.74	80.00 ± 42.16 80.00 ± 42.16	100.00 ± 0.00 100.00 ± 0.00
OR2	88.28 ± 1.64 86.08 ± 2.51	88.93 ± 1.22 85.50 ± 3.42	76.92 ± 4.90 74.41 ± 5.45	81.60 ± 4.23 77.28 ± 3.76
PB	98.70 ± 1.86 99.06 ± 0.64	95.90 ± 1.12 97.86 ± 0.67	69.77 ± 2.62 68.75 ± 2.63	67.19 ± 3.17 66.63 ± 4.09
Acc	62.05 ± 3.31 60.02 ± 7.55	65.02 ± 4.79 66.36 ± 4.42	20.05 ± 27.57 21.95 ± 29.61	20.46 ± 28.56 23.43 ± 31.40
Ecoli	90.32 ± 1.26 89.29 ± 1.27	90.80 ± 2.59 88.04 ± 4.44	84.41 ± 6.13 84.92 ± 6.82	85.92 ± 6.69 86.05 ± 6.55
SSMCR	87.93 ± 0.93 87.93 ± 0.93	87.63 ± 1.29 87.63 ± 1.29	85.71 ± 1.98 84.97 ± 2.03	87.42 ± 1.37 86.79 ± 1.50
UMD	98.42 ± 1.56 98.97 ± 1.50	100.00 ± 0.00 99.58 ± 0.88	100.00 ± 0.00 100.00 ± 0.00	100.00 ± 0.00 100.00 ± 0.00
RD	83.07± 2.33 80.87 ± 5.27	82.45 ± 2.18 77.63 ± 3.77	70.61 ± 12.89 72.92 ± 14.53	71.29 ± 13.63 73.12 ± 12.83

5.8 Universal Consistency of CPB-GLPUAL

5.8.1 Introduction to Bayes Risk

Firstly we define the 0-1 risk, i.e., the expected error rate, of a binary classifier with decision function $f^*(\boldsymbol{x}) = \mathrm{sgn}(f(x)) \in \{-1, 1\}$ as:

$$\mathscr{R}_{0-1}(f^*) = \int_{(\boldsymbol{x},y)\in\mathscr{S}} \beth(f^*(\boldsymbol{x}) \neq y) P(\boldsymbol{X} = \boldsymbol{x}, Y = y) d\boldsymbol{x} dy = P[f^*(\boldsymbol{X}) \neq Y], \quad (5.44)$$

where $\mathscr{S}$ is the domain of the instance $(\boldsymbol{X}, Y)$ and $\beth(\cdot)$ is the indicator function. $\mathscr{R}_{0-1}$ indicates the probability of a classifier to misclassify instance $(\boldsymbol{X}, Y)$ selected at random from this domain.

Let $\mathscr{S}_x$ be the domain of $\boldsymbol{X}$. We can divide $\mathscr{S}_x$ into the following three regions by the class which instance $(\boldsymbol{X} = \boldsymbol{x}, Y = y)$ is more likely to belong to:

$$\begin{aligned}
\boldsymbol{Z}_+ &= \{\boldsymbol{x} \in \mathscr{S}_x : P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) > P(Y = -1 \mid \boldsymbol{X} = \boldsymbol{x})\}, \\
\boldsymbol{Z}_- &= \{\boldsymbol{x} \in \mathscr{S}_x : P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) < P(Y = -1 \mid \boldsymbol{X} = \boldsymbol{x})\}, \\
\boldsymbol{Z}_0 &= \{\boldsymbol{x} \in \mathscr{S}_x : P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = P(Y = -1 \mid \boldsymbol{X} = \boldsymbol{x})\}.
\end{aligned} \quad (5.45)$$

Based on the three regions $\boldsymbol{Z}_+$, $\boldsymbol{Z}_-$ and $\boldsymbol{Z}_0$ in Equation 5.45, the Bayes decision function can be defined as

$$f^*_{\text{Bayes}} = \begin{cases} 1, & \boldsymbol{x} \in \boldsymbol{Z}_+ \cup \boldsymbol{Z}_0; \\ -1, & \boldsymbol{x} \in \boldsymbol{Z}_-. \end{cases} \quad (5.46)$$

The classifier with the Bayes decision function is called the Bayes classifier and the 0-1 risk of the Bayes classifier is termed as Bayes risk. According to [74], the Bayes risk can be transformed to the following form from Equation 5.44:

$$\mathscr{R}_{\text{Bayes}} = \mathscr{R}_{0-1}(f^*_{\text{Bayes}}) = \int_{\boldsymbol{x}\in\mathscr{S}_x} \eta(\boldsymbol{x}) P(\boldsymbol{X} = \boldsymbol{x}) d\boldsymbol{x}, \quad (5.47)$$

where

$$\eta(\boldsymbol{x}) = \begin{cases} P(Y = -1|\boldsymbol{X} = \boldsymbol{x}), & \boldsymbol{x} \in \mathbf{Z}_+ \cup \mathbf{Z}_0; \\ P(Y = 1|\boldsymbol{X} = \boldsymbol{x}), & \boldsymbol{x} \in \mathbf{Z}_-. \end{cases} \tag{5.48}$$

The Bayes classifier is the optimal classifier for the lowest 0-1 risk [75] i.e., for any classifier with decision function f^*, the following relation holds:

$$\mathscr{R}_{0-1}(f^*) \geq \mathscr{R}_{\text{Bayes}}. \tag{5.49}$$

5.8.2 Universal Consistency of CPB-GLPUAL

Suppose that the feature mapping $\boldsymbol{\phi}(\cdot)$ is used to train CPB-GLPUAL and define the covering number $\mathscr{N}\left(\left(\mathscr{S}_x, d_{\boldsymbol{\phi}}\right), \varepsilon\right)$, where metric $d_{\boldsymbol{\phi}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_i)\|_2^2$, to be the minimum amount of hyper-spheres with diameter $\varepsilon > 0$ to cover the entire metric space $(\mathscr{S}_x, d_{\boldsymbol{\phi}})$. Then according to [76], the universal kernel $\boldsymbol{\Phi}^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\phi}(\boldsymbol{x}_1)^T \boldsymbol{\phi}(\boldsymbol{x}_2)$ specifies the kernel functions satisfying the following two conditions:

- $\boldsymbol{\phi}(\cdot)$ is continuous.
- $\forall \varepsilon > 0$, $\mathscr{N}\left(\left(\mathscr{S}_x, d_{\boldsymbol{\phi}}\right), \varepsilon\right)$ can be regarded as a finite function w.r.t. ε.

In practice, the Bayes classifier is not able to be obtained due to the unknown distribution of the ground-truth population. In this case, people may expect the proposed classifier to be an approximation of the Bayes classifier. Therefore, the universal consistency is introduced to measure the gap between the Bayes risk and the 0-1 risk of the proposed classifier with the size of dataset increasing [77] with data $\boldsymbol{X}_{[pu]}$ to follow any distribution.

Define $c' = \frac{2c}{\lambda}$ and the decision function of CPB-GLPUAL trained from sample size n_{pu} to be $f_{cpb}^{*n_{pu}}$. Specific to CPB-GLPUAL, the universal consistency of CPB-GLPUAL can be summarised into the following theorem:

Theorem 3 *Firstly, suppose that $\mathscr{S}_x$ is compact, and $\boldsymbol{\Phi}^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\phi}(\boldsymbol{x}_1)^T \boldsymbol{\phi}(\boldsymbol{x}_2)$ is a universal kernel function. Secondly, suppose that there exists constant $\alpha > 0$ to* *satisfy* $\mathscr{N}\left(\left(\mathscr{S}_x, d_{\phi}\right), \varepsilon\right) \in \mathscr{O}\left(\varepsilon^{-\alpha}\right)$*. Thirdly, suppose that there exists constant* δ *satisfying* $0 < \delta < \frac{1}{\alpha}$*, and when the sample size* n_{pu} *tends to infinity, the value of* c' *also tends to infinity with* $c' \in \mathscr{O}(n_{pu}^{\delta})$*. In this case,* $\forall \varepsilon > 0$*, we have*

$$P^{n_{pu}}\left[\mathscr{R}_{0-1}\left(f_{cpb}^{*n_{pu}}\right) - \mathscr{R}_{Bayes} \leqslant \varepsilon\right] \to 1,$$

where $\mathscr{R}_{0-1}\left(f_{cpb}^{*n_{pu}}\right)$ *is the 0-1 risk of the trained decision function* $f_{cpb}^{*n_{pu}}$ *of CPB-GLPUAL at sample size* n_{pu}*.*

Theorem 3 indicates that with the size of the training set increasing, the gap between Bayes risk and the 0-1 risk of GLPUAL tends to 0 by probability.

In the field of PU learning, [23, 26, 78, 79] managed to show the similar theorem for their proposed PU classifiers. However, the proof in [23, 26, 78, 79] cannot hold once the regularised terms for the model parameters are added into the objective function while the L2-norm regularised terms is considered in our proof. Moreover, it should be noted that [23, 26, 78] force the loss in their objective functions to be the 0-1 loss or its estimate and [26] makes additional assumptions on the distribution of data $\boldsymbol{X}_{[pu]}$.

In CPB-GLPUAL, we have an L2-norm regularisation in the objective function, hence the proof in [23, 26, 78, 79] cannot be used here, and we developed our own proof. In our proof, the asymmetric loss in the objective function does not need to be the estimate of the 0-1 loss. Furthermore there is only one weak assumption related to the distribution of $\boldsymbol{X}_{[pu]}$ made in Section 5.8.3.1. This indicates that the universal consistency of CPB-GLPUAL has broader scope of application and we have more choices on the loss function in future works.

5.8.3 An Approximate of CPB-GLPUAL to Simplify the Proof of Universal Consistency

As the initial exploration of the universal consistency on PU classifiers, we firstly construct a PN classifier to approximate CPB-GLPUAL and then construct a lower probabilistic boundary for the gap between the 0-1 risk of this PN classifier and the Bayes risk.

5.8.3.1 Construction of the Approximate

According to [21], the local constraint $(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)^T\boldsymbol{R}(\boldsymbol{X}_{[pu]}\boldsymbol{\beta}+\mathbf{1}_{pu}\beta_0)$ in the objective function of GLPUAL in Equation 5.8 can be transformed to

$$\frac{2}{n_{pu}}\sum_{\boldsymbol{x}_i,\boldsymbol{x}_j\text{are the KNN of each other}}\exp\left(-\sigma^{-1}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2\right)\left(f(\boldsymbol{\phi}(\boldsymbol{x}_i))-f(\boldsymbol{\phi}(\boldsymbol{x}_j))\right)^2, \tag{5.50}$$

where $f(\boldsymbol{\phi}(\boldsymbol{x}))=\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}+\beta_0$.

Define the r.v. $\mathscr{X}_{k[i]}^{[mx]}$ to be the kth max value of set $\{(-\|\boldsymbol{X}_i-\boldsymbol{X}_j\|_2^2):j=1,2,\ldots,i-1,i+1,\ldots,n_{pu}\}$. When $k=1$, the cdf of $\mathscr{X}_{1[i]}^{[mx]}$ is

$$P[\mathscr{X}_{1[i]}^{[mx]}\leq x]=\mathscr{P}_{i-}^{n_{pu}-1}(x), \tag{5.51}$$

where $\mathscr{P}_{i-}(x)$ is the cdf of the r.v. $-\|\boldsymbol{X}_i-\boldsymbol{X}_j\|_2^2$ for $j=1,2,\ldots,i-1,i+1,\ldots,n_{pu}$.

Here we make an assumption that $\forall x<0$, there is $\mathscr{P}_{i-}(x)<1$. In this case, as n_{pu} tends to infinity, $P[\mathscr{X}_{1[i]}^{[mx]}\leq x]$ tends to 0. Therefore, it is obvious to find only the pdf $p[\mathscr{X}_{1[i]}^{[mx]}=0]$ tends to infinity as n_{pu} tends to infinity, so that $\mathscr{X}_{1[i]}^{[mx]}$ converges to 0 in probability. Then for $k=2$, we have

$$P[\mathscr{X}_{2[i]}^{[mx]}\leq x]=\mathscr{P}_{i-}^{n_{pu}-1}(x)+(1-P[\mathscr{X}_{1[i]}^{[mx]}\leq x])\mathscr{P}_{i-}^{n_{pu}-2}(x). \tag{5.52}$$

We can obtain similar conclusion that $\mathscr{X}_{2[i]}^{[mx]}$ converges to 0 in probability as n_{pu} tends to infinity. Following this way, we have

$$P[\mathscr{X}_{k[i]}^{[mx]}\leq x]=P[\mathscr{X}_{1[i]}^{[mx]}\leq x]+\sum_{a=2}^{k}(1-P[\mathscr{X}_{a[i]}^{[mx]}\leq x])\mathscr{P}_{i-}^{n_{pu}-a}(x). \tag{5.53}$$

Thus, for limited k, only $p[\mathscr{X}_{k[i]}^{[mx]}=0]$ tends to infinity as n_{pu} tends to infinity and $\mathscr{X}_{k[i]}^{[mx]}$ converges to 0 in probability. Furthermore, the case appearing with

$\mathscr{X}_{k[i]}^{[mx]} = 0$ is continuous function $(f(\boldsymbol{\phi}(\boldsymbol{x}_i)) - f(\boldsymbol{\phi}(\boldsymbol{x}_j)))^2 = 0$ for $\boldsymbol{x}_j$ to be the kth NN of $\boldsymbol{x}_i$. Hence the local constraint in Equation 5.50 can be regarded as the weighted average of the r.v.s converging to 0 in probability. Therefore, the local constraint also converges to 0 in probability as n_{pu} tends to infinity.

In this case, we only need to consider the weighted average of the losses and the regularised term in the objective function of CPB-GLPUAL for sufficient large n_{pu}, which converges to the following PN objective function in probability with kernel trick applied according to [23]:

$$\boldsymbol{\beta}^T\boldsymbol{\beta} + \frac{c'}{n_{pu}} \sum_{i=1}^{n_{pu}} l(f(\boldsymbol{\phi}(\boldsymbol{x}_i); \boldsymbol{\beta}, \beta_0), y_i) \tag{5.54}$$

where $c' = \frac{2c}{\lambda}$ the asymmetric loss function is

$$l(f(\boldsymbol{x}; \boldsymbol{\beta}, \beta_0), y) = \begin{cases} [1 - \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta} - \beta_0]_+, y = 1; \\ |1 + \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta} + \beta_0|, y = -1. \end{cases} \tag{5.55}$$

The predictive score function of this approximate of CPB-GLPUAL is also

$$f(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta} + \beta_0.$$

In this case, there is $\mathscr{R}_{0-1}\left(f_{\text{cpb}}^{*n_{pu}}\right) \to \mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)$ with n_{pu} increasing, where $\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)$ is the 0-1 risk of the trained decision function $f_{\text{ap}}^{*n_{pu}}$ of the approximate of CPB-GLPUAL in Equation 5.54 with $c' = c_{n_{pu}}$ and the sample size n_{pu}.

Furthermore, the objective function of the approximate of CPB-GLPUAL in Equation 5.54 is equivalent to the following constrained form for the optimisation with slack variable $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{n_{pu}})^T \in \mathbb{R}^{n_{pu} \times 1}$ introduced:

$$\begin{aligned}
&\boldsymbol{\beta}^T \boldsymbol{\beta} + \frac{c'}{n_{pu}} \sum_{i=1}^{n_{pu}} \xi_i \\
s.t.\ &\xi_i \geq 1 - \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\beta} - \beta_0, i = 1, \ldots, n_{pu} - n_u; \\
&\xi_i \geq 0, i = 1, \ldots, n_{pu} - n_u; \\
&\xi_i \geq 1 + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\beta} + \beta_0, i = n_{pu} - n_u + 1, \ldots, n_{pu}; \\
&\xi_i \geq -1 - \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\beta} - \beta_0, i = n_{pu} - n_u + 1, \ldots, n_{pu}.
\end{aligned} \tag{5.56}$$

In this case, one can find the kernel form of the optimisation of Equation 5.56 via the KKT condition w.r.t. $\boldsymbol{\beta}$ as:

$$\begin{aligned}
&\min_{\boldsymbol{\upsilon}, \beta_0, \boldsymbol{\xi}} \frac{1}{2} \boldsymbol{\upsilon}^T \boldsymbol{\Phi}^*(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]}) \boldsymbol{\upsilon} + \frac{c'}{n_{pu}} \sum_{i=1}^{n_{pu}} \xi_i \\
s.t.\ &\xi_i \geq 1 - \boldsymbol{\Phi}^*(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]}) \boldsymbol{\upsilon} - \beta_0, i = 1, \ldots, n_{pu} - n_u; \\
&\xi_i \geq 0, i = 1, \ldots, n_{pu} - n_u; \\
&\xi_i \geq 1 + \boldsymbol{\Phi}^*(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]}) \boldsymbol{\upsilon} + \beta_0, i = n_{pu} - n_u + 1, \ldots, n_{pu}; \\
&\xi_i \geq -1 - \boldsymbol{\Phi}^*(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]}) \boldsymbol{\upsilon} - \beta_0, i = n_{pu} - n_u + 1, \ldots, n_{pu}.
\end{aligned} \tag{5.57}$$

where the (i, j) element of kernel matrix $\boldsymbol{\Phi}^*(\boldsymbol{X}_{[pu]}, \boldsymbol{X}_{[pu]})$ is $\boldsymbol{\Phi}^*(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

5.8.3.2 Universal Consistency of the Approximate

To prove Theorem 3, firstly we proved the universal consistency of the approximate of CPB-GLPUAL with the objective function in the form in Equation 5.56 based on the idea in [80], which proved the universal consistency of the classic supervised SVM. We can give the following Theorem 4 for the approximate of CPB-GLPUAL:

Theorem 4 *Suppose $\mathscr{S}_x$ is compact and the kernel function $\Phi(\cdot)$ is universal. $\forall 0 < \varepsilon < 1$, we can find a constant $c^* > 0$ such that for all $c' \geq c^*$ there is*

$$\mathrm{P}^{n_{pu}} \left[\mathscr{R}_{0-1} \left(f_{ap}^{*n_{pu}} \right) - \mathscr{R}_{Bayes} \leq \varepsilon \right] \geq 1 - 2Me^{-\frac{\varepsilon^6 n_{pu}}{2^{29} M^2}},$$

where $M = \frac{64}{\varepsilon} \mathscr{N} \left((\mathscr{S}_x, d_\Phi), \frac{\varepsilon}{32\sqrt{c'}} \right)$.

5.8.4 Three Steps of the Proof of Theorem 3

The proof of Theorem 3 is based on the proof of Theorem 4, by the following three steps:

1. Pick the 'representative' instances $\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n$ form the domain $\mathscr{S}_x$ to satisfy $\mathrm{P}[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n] \geq 1-2Me^{-\frac{\varepsilon^6 n_{pu}}{2^{29}M^2}}$.

2. Show that once $\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n$ are the 'representative' instances, $\mathscr{R}_{0-1}\left(f_{\mathrm{ap}}^{*n_{pu}}\right) - \mathscr{R}_{\mathrm{Bayes}} \leq \varepsilon$ by proof by contradiction. In this case Theorem 4 can be proved.

3. Show $\mathscr{R}_{0-1}\left(f_{\mathrm{cpb}}^{*n_{pu}}\right) \to \mathscr{R}_{0-1}\left(f_{\mathrm{ap}}^{*n_{pu}}\right) \to \mathscr{R}_{\mathrm{Bayes}}$. In this case Theorem 3 can be proved based on Theorem 4.

5.8.5 Step 1: Construction of the 'Representative' Dataset

The way to achieve Step 1 is based on the domain $\mathscr{S}_x$ itself, which is independent of the objective function, the loss function and the predictive score function. Therefore, what we need to do is completely the same as the corresponding part in [80]. In this case, we summarise the important details of [80] in this section with the proof (referring to the proof of Lemma 2 to Lemma 4 in [80]) skipped and then add some additional analysis.

The key idea of the construction of the 'representative' dataset is to sample certain amount positive instances and negative instances from many small subsets of domain $\mathscr{S}_x$. In this case, by recalling $\eta(\boldsymbol{x})$ defined in Equation 5.48, firstly we can divide $\mathscr{S}_x$ into the following subsets:

$$\mathscr{S}_{x[i]} = \begin{cases} \{\boldsymbol{x} \in \mathscr{S}_x : i2^{-\rho} \leq \eta(\boldsymbol{x}) < (i+1)2^{-\rho}\}, & i = 0,1,\dots,2^{\rho-1}-2, \\ \{\boldsymbol{x} \in \mathscr{S}_x : i2^{-\rho} \leq \eta(\boldsymbol{x}) \leq \frac{1}{2}\} & i = 2^{\rho-1}-1. \end{cases} \tag{5.58}$$

where ρ is the integer meeting $2^{-\rho} \leq \tau \leq 2^{-\rho+1}$ and $\tau = \varepsilon/32$; this leads to the following relationship:

$$\sum_{i=0}^{2^{\rho-1}-1} \frac{i}{2^\rho} P[\boldsymbol{X} \in \mathscr{S}_{x[i]}] \leq \mathscr{R}_{0-1} \leq \sum_{i=0}^{2^{\rho-1}-1} \frac{i}{2^\rho} P[\boldsymbol{X} \in \mathscr{S}_{x[i]}] + \frac{1}{2^\rho} \sum_{i=0}^{2^{\rho-1}-1} P[\boldsymbol{X} \in \mathscr{S}_{x[i]}]$$
$$\leq \sum_{i=0}^{2^{\rho-1}-1} \frac{i}{2^\rho} P[\boldsymbol{X} \in \mathscr{S}_{x[i]}] + \tau, \tag{5.59}$$

where 0-1 risk $\mathscr{R}_{0-1}$ is defined in Equation 5.44.

To control the amount of the positive and negative instance in the 'representative' dataset, we need to divide $\mathscr{S}_{x[i]}, i = 0,1,\ldots,2^{\rho-1}-2$, into $\mathscr{S}^1_{x[i]} = \mathscr{S}_{x[i]} \cap \boldsymbol{Z}_+$ and $\mathscr{S}^{-1}_{x[i]} = \mathscr{S}_{x[i]} \cap \boldsymbol{Z}_-$. Furthermore, we can construct a 'large' enough compact subset $\mathscr{B}^j_{[i]}$ of $\mathscr{S}^j_{x[i]}$, i.e.,

$$P\left[\boldsymbol{X} \in \mathscr{S}^j_{x[i]} \backslash \mathscr{B}^j_{[i]}\right] \leq \tau 2^{-\rho}, \;\; i = 0,\ldots,2^{\rho-1}-2, j \in \{-1,1\}. \tag{5.60}$$

Furthermore, there exists subset $\mathscr{B}_{[2^{\rho-1}-1]}$ of $\mathscr{S}_{x[2^{\rho-1}-1]}$ meeting

$$P\left[\boldsymbol{X} \in \mathscr{S}_{x[2^{\rho-1}-1]} \backslash \mathscr{B}_{[2^{\rho-1}-1]}\right] \leq \tau 2^{-\rho} \tag{5.61}$$

For convenience, let $\mathscr{B}^1_{[2^{\rho-1}-1]} = \mathscr{B}_{[2^{\rho-1}-1]} \cap (\boldsymbol{Z}_+ \cup \boldsymbol{Z}_0)$ and $\mathscr{B}^{-1}_{[2^{\rho-1}-1]} = \mathscr{B}_{[2^{\rho-1}-1]} \cap \boldsymbol{Z}_-$.

As proved in Lemma 2 of [80], when $\boldsymbol{\Phi}^*(\boldsymbol{X}_1,\boldsymbol{X}_2) = \boldsymbol{\phi}(\boldsymbol{X}_1)\boldsymbol{\phi}(\boldsymbol{X}_2)^T$ to be universal kernel, there exists value $\tilde{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ to satisfy:

$$\begin{aligned}
&\boldsymbol{\phi}(\boldsymbol{x})^T\tilde{\boldsymbol{\beta}} \in [1,1+\tau],\; \boldsymbol{x} \in \cup_{i=0}^{2^{\rho-1}-2} \mathscr{B}^1_{[i]},\\
&\boldsymbol{\phi}(\boldsymbol{x})^T\tilde{\boldsymbol{\beta}} \in [-(1+\tau),-1],\; \boldsymbol{x} \in \cup_{i=0}^{2^{\rho-1}-2} \mathscr{B}^{-1}_{[i]},\\
&\boldsymbol{\phi}(\boldsymbol{x})^T\tilde{\boldsymbol{\beta}} \in [-\tau,\tau],\; \boldsymbol{x} \in \mathscr{B}_{[2^{\rho-1}-1]}\\
&\boldsymbol{\phi}(\boldsymbol{x})^T\tilde{\boldsymbol{\beta}} \in [-(1+\tau),1+\tau],\; \boldsymbol{x} \notin \cup_{j=-1,1} \cup_{i=0}^{2^{\rho-1}-1} \mathscr{B}^j_{[i]}.
\end{aligned} \tag{5.62}$$

Equation 5.62 is used to construct the upper bound of the contradiction in Section 5.8.6.1.

Let $\sigma = \tau/\sqrt{c'}$. For $i = 0, \dots, 2^{\rho-1}-1$ and $j = -1, 1$ we are able to divide $\mathscr{B}^j_{[i]}$ into finite partition $\tilde{\mathbb{A}}^j_i$ with the diameter of each set $\mathscr{A} \in \tilde{\mathbb{A}}^j_i$ no greater than σ in the kernel space. According to the definition of the covering numbers, the cardinality of $\tilde{\mathbb{A}}^j_i$ is no greater than $\mathscr{N}\left(\left(\mathscr{S}_x, d_{\boldsymbol{\phi}}\right), \sigma\right)$. Based on this, we can define

$$\mathbb{A}^j_i = \left\{\mathscr{A} \in \tilde{\mathbb{A}}^j_i : P[\boldsymbol{X} \in \mathscr{A}] \geq \frac{2\tau}{M}\right\}, \tag{5.63}$$

with $2^\rho \leq |\cup_{j=-1,1} \cup_{i=0}^{2^{\rho-1}-1} \mathbb{A}^j_i| \leq M$. Therefore, recalling $M = \frac{64}{\varepsilon}\mathscr{N}\left((\mathscr{S}_x, d_\Phi), \frac{\varepsilon}{32\sqrt{c'}}\right)$, there is

$$\begin{aligned}\sum_{\mathscr{A} \in \mathbb{A}^j_i} P[\boldsymbol{X} \in \mathscr{A}] &= P[\boldsymbol{X} \in \mathscr{B}^j_{[i]}] - P[\boldsymbol{X} \in \mathscr{B}^j_{[i]} \setminus \cup_{\mathscr{A} \in \mathbb{A}^j_i} \mathscr{A}] \\ &\geq P[\boldsymbol{X} \in \mathscr{B}^j_{[i]}] - \frac{2\tau}{M}\mathscr{N}\left(\left(\mathscr{S}_x, d_{\boldsymbol{\phi}}\right), \sigma\right) \\ &= P[\boldsymbol{X} \in \mathscr{B}^j_{[i]}] - \frac{2\tau}{M}\frac{\tau}{2}M \\ &= P[\boldsymbol{X} \in \mathscr{B}^j_{[i]}] - \tau^2 \geq P[\boldsymbol{X} \in \mathscr{B}^j_{[i]}] - \tau.\end{aligned} \tag{5.64}$$

For convenience, let $\mathscr{B}^{*j}_{[i]} = \cup_{\mathscr{A} \in \mathbb{A}^j_i} \mathscr{A}$ for $i = 0, \dots, 2^{\rho-1}-1, j \in \{-1, 1\}$.

Consider the following conditions for the dataset $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})\}$ with $n_{pu} \gg 2^{\rho+1}$:

$$\begin{aligned} F^+_{n_{pu},\mathscr{A}} &= \left\{((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})) : |\{l : \boldsymbol{x}_l \in \mathscr{A}, y_l = j\}| \geq n_{pu}(1-\tau)\left(1 - \frac{i+1}{2^\rho}\right)P[\boldsymbol{X} \in \mathscr{A}]\right\}, \\ F^-_{n_{pu},\mathscr{A}} &= \left\{((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})) : |\{i : \boldsymbol{x}_l \in \mathscr{A}, y_i \neq j\}| \geq n_{pu}(1-\tau)\frac{i}{2^\rho}P[\boldsymbol{X} \in \mathscr{A}]\right\}, \end{aligned} \tag{5.65}$$

where $i = 0, \dots, 2^{\rho-1}-2, j \in \{-1, 1\}$ and $\mathscr{A} \in \mathbb{A}^j_i$. Besides, for $\mathscr{A} \in \mathbb{A}^j_{2^{\rho-1}-1}, j \in \{-1, 1\}$ we can define the conditions as:

$$\begin{aligned} F^+_{n_{pu},\mathscr{A}} &= \left\{((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})) : |\{l : \boldsymbol{x}_l \in \mathscr{A}, y = j\}| \geq n_{pu}(1-\tau)\left(\frac{1}{2} - \frac{1}{2^\rho}\right)P[\boldsymbol{X} \in \mathscr{A}]\right\}. \\ F^-_{n_{pu},\mathscr{A}} &= \left\{((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})) : |\{l : \boldsymbol{x}_l \in \mathscr{A}, y = -1\}| \geq n_{pu}(1-\tau)\left(\frac{1}{2} - \frac{1}{2^\rho}\right)P[\boldsymbol{X} \in \mathscr{A}].\right\} \end{aligned} \tag{5.66}$$

Let $F_{n_{pu}} = \cap_{j \in \{-1,1\}} \cap_{i=0}^{2^{\rho-1}-1} \cap_{\mathscr{A} \in \mathbb{A}^j_i}\left(F^+_{n_{pu},\mathscr{A}} \cap F^-_{n_{pu},\mathscr{A}}\right)$. We can construct the 'representative' dataset by making the dataset meet the above conditions in Equation 5.65 and Equation 5.66, i.e., $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})\} \in F_{n_{pu}}$. The probability of obtaining such ’representative’ dataset via i.i.d. sampling form the population is

$$\begin{aligned} P^{n_{pu}}\left(F_{n_{pu}}\right) &\geq 1-2Me^{-2\left(\tau^6/M^2\right)n_{pu}} \\ &= 1-2Me^{-\frac{\varepsilon^6 n_{pu}}{2^{29}M^2}}, \end{aligned} \tag{5.67}$$

for $n_{pu} \gg 2^{\rho+1}$ as proved in Lemma 3 of [80]. Besides, there are at least 2^ρ positive instances and at least 2^ρ negative instances in the ’representative’ dataset since $P[\boldsymbol{X} \in \mathscr{A}]$ in Equation 5.65 and Equation 5.66 is always greater than 0 according to Equation 5.63,

Additionally, let $\mathscr{E}_i^j$ denote the subset of $\mathscr{S}_{x[i]}^j$ where the proposed classifier and the Bayes classifier output different class results, i.e.,

$$\mathscr{E}_i^j = \left\{\boldsymbol{x} \in \mathscr{S}_{x[i]}^j : \operatorname{sgn}(f_{\text{ap}}^{n_{pu}}(x)) \neq j\right\}, \tag{5.68}$$

where $i = 0,\ldots,2^{\rho-1}-2, j \in \{-1,1\}$. Also, for $i = 0,\ldots,2^{\rho-1}-1$ and $j = 1$, there is

$$\mathscr{E}_{2^{\rho-1}-1}^1 = \left\{x \in \mathscr{S}_{x[2^{\rho-1}-1]} \cap \mathbf{Z}_+ : \operatorname{sgn}(f_{\text{ap}}^{n_{pu}}(x)) \neq 1\right\}. \tag{5.69}$$

For $i = 0,\ldots,2^{\rho-1}-1$ and $j = -1$, there is

$$\mathscr{E}_{2^{\rho-1}-1}^{-1} = \left\{x \in \mathscr{S}_{x[2^{\rho-1}-1]} \cap \mathbf{Z}_- : \operatorname{sgn}(f_{\text{ap}}^{n_{pu}}(x)) \neq -1\right\}. \tag{5.70}$$

According to Lemma 4 in [80] and the assumption in Inequality 5.72 we can obtain

$$\varepsilon - 2\tau < \sum_{i=0}^{2^{\rho-1}-2} \left(1-\frac{i}{2^{\rho-1}}\right) P\left[\boldsymbol{X} \in \mathscr{E}_i^1 \cup \mathscr{E}_i^{-1}\right]. \tag{5.71}$$

5.8.6 Step 2: Proof of Theorem 4 by Contradiction

In this section, we prove that once $\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\dots,(\boldsymbol{x}_{n_{pu}},y_{n_{pu}})\}$ are the 'representative' instances, we will have $\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)-\mathscr{R}_{\text{Bayes}}<\varepsilon$ via the proof by contradiction in an inequality. In this case Theorem 4 can be proved. More specifically, firstly we construct the upper bound of the inequality in Section 5.8.6.1. Secondly, we construct the lower bound of the inequality in Section 5.8.6.2. Thirdly we prove the lower bound is larger than the upper bound of the inequality to cause contradiction in Section 5.8.6.3.

5.8.6.1 Upper Bound of the Inequality for Contradiction

Firstly, assume that there is a 'representative' dataset $\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\dots,(\boldsymbol{x}_{n_{pu}},y_{n_{pu}})\}\in F_n$ with

$$\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)-\mathscr{R}_{\text{Bayes}}>\varepsilon. \tag{5.72}$$

Define $\boldsymbol{\beta}_{[\text{ap}]}$ and $\beta_{[\text{ap}]0}$ to be the optimal solution of the objective function in Equation 5.56. Then let the value of the slack variable of instance $(\boldsymbol{x}_l,y_l)$ in Equation 5.56 of instance $(\boldsymbol{x}_l,y_l)$, with $\boldsymbol{\beta}=\boldsymbol{\beta}_{[\text{ap}]}$ and $\beta_0=\beta_{[\text{ap}]0}$, to be $\xi_{[\text{ap}]l}$. Similarly, let the value of the slack variable of instance $(\boldsymbol{x}_l,y_l)$, with $\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}$ and $\beta_0=0$, to be $\tilde{\xi}_l$.

Furthermore, according to the relationships in Equation 5.62 and the constraints in Equation 5.56, there are the following eight scenarios for $\tilde{\xi}_l$:

- For $x_l\in\mathscr{B}_{[i]}^{*1},i=0,1,\dots,2^{\rho-1}-2$, and $y_l=1$, there is $1-\boldsymbol{\phi}(\boldsymbol{x}_l)^T\tilde{\boldsymbol{\beta}}\in[-\tau,0]$ thus in this case we can let $\tilde{\xi}_l=0$;

- For $x_l\in\mathscr{B}_{[i]}^{*1},i=0,1,\dots,2^{\rho-1}-2$, and $y_l=-1$, there is $1+\boldsymbol{\phi}(\boldsymbol{x}_l)^T\tilde{\boldsymbol{\beta}}\in[2,2+\tau]$ thus in this case we can let $\tilde{\xi}_l=2+\tau$;

- For $x_l\in\mathscr{B}_{[i]}^{*-1},i=0,1,\dots,2^{\rho-1}-2$, and $y_l=1$, there is $1-\boldsymbol{\phi}(\boldsymbol{x}_l)^T\tilde{\boldsymbol{\beta}}\in[2,2+\tau]$ thus in this case we can also let $\tilde{\xi}_l=2+\tau$;

- For $x_l\in\mathscr{B}_{[i]}^{*-1},i=0,1,\dots,2^{\rho-1}-2$, and $y_l=-1$, there is $1+\boldsymbol{\phi}(\boldsymbol{x}_l)^T\tilde{\boldsymbol{\beta}}\in[-\tau,0]$ thus in this case we can let $\tilde{\xi}_l=\tau$;

- For $x_l \in \mathscr{B}_{[2^{\rho-1}-1]}$ and $y_l = 1$, there is $1 - \boldsymbol{\phi}(\boldsymbol{x}_l)^T \tilde{\boldsymbol{\beta}} \in [1-\tau, 1+\tau]$ thus in this case we can let $\tilde{\xi}_l = 1+\tau$;

- For $x_l \in \mathscr{B}_{[2^{\rho-1}-1]}$ and $y_l = -1$, there is $1 + \boldsymbol{\phi}(\boldsymbol{x}_l)^T \tilde{\boldsymbol{\beta}} \in [1-\tau, 1+\tau]$ thus in this case we can also let $\tilde{\xi}_l = 1+\tau$;

- For $x_l \notin (\cup_{i=0}^{2^{\rho-1}-2} \cup_{j=\{-1,1\}} \mathscr{B}_{[i]}^{*j}) \cup \mathscr{B}_{[2^{\rho-1}-1]}$ and $y_l = 1$ there is $1 - \boldsymbol{\phi}(\boldsymbol{x}_l)^T \tilde{\boldsymbol{\beta}} \in [-\tau, 2+\tau]$ thus in this case we can let $\tilde{\xi}_l = 2+\tau$;

- For $x_l \notin (\cup_{i=0}^{2^{\rho-1}-2} \cup_{j=\{-1,1\}} \mathscr{B}_{[i]}^{*j}) \cup \mathscr{B}_{[2^{\rho-1}-1]}$ and $y_l = -1$ there is $1 + \boldsymbol{\phi}(\boldsymbol{x}_l)^T \tilde{\boldsymbol{\beta}} \in [-\tau, 2+\tau]$ thus in this case we can also let $\tilde{\xi}_l = 2+\tau$.

Then let $n_1, n_1^+, n_1^-, n_2, n_3, n_4$ denote the number of specific instances in the 'representative' set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_{n_{pu}}, y_{n_{pu}})\}$ as:

$$\begin{aligned}
& n_1^+ = \left|\left\{l : \boldsymbol{x}_l \in \cup_{i=0}^{2^{\rho-1}} \mathscr{B}_{[i]}^{*1}, y_l = 1\right\}\right|, n_1^- = \left|\left\{l : \boldsymbol{x}_l \in \cup_{i=0}^{2^{\rho-1}} \mathscr{B}_{[i]}^{*-1}, y_l = -1\right\}\right|, \\
& n_1 = n_1^+ + n_1^-, \\
& n_2 = \left|\left\{l : \boldsymbol{x}_l \in \cup_{i=0}^{2^{\rho-1}} \mathscr{B}_{[i]}^{*-1}, y_l = 1\right\}\right| + \left|\left\{l : \boldsymbol{x}_l \in \cup_{i=0}^{2^{\rho-1}-2} \mathscr{B}_{[i]}^{*1}, y_l = -1\right\}\right|, \\
& n_3 = \left|\left\{l : \boldsymbol{x}_l \in \mathscr{B}_{[2^{\rho-1}-1]}\right\}\right|, \\
& n_4 = \left|\left\{l : \boldsymbol{x}_l \notin (\cup_{i=0}^{2^{\rho-1}-2} \cup_{j=\{-1,1\}} \mathscr{B}_{[i]}^{*j}) \cup \mathscr{B}_{[2^{\rho-1}-1]}\right\}\right|.
\end{aligned} \tag{5.73}$$

According to Equation 5.73, obviously there is $n_{pu} = n_1 + n_2 + n_3 + n_4$. Furthermore as $(\boldsymbol{\beta}_{[\text{ap}]}, \beta_{[\text{ap}]0})$ is the optimal solution of $(\boldsymbol{\beta}, \beta_0)$, there is

$$\begin{aligned}
\boldsymbol{\beta}_{[\text{ap}]}^T \boldsymbol{\beta}_{[\text{ap}]} + \frac{c'}{n_{pu}} \sum_{l=1}^{n_{pu}} \xi_{[\text{ap}]l} \leq \tilde{\boldsymbol{\beta}}^T \tilde{\boldsymbol{\beta}} + \frac{c'}{n_{pu}} \sum_{l=1}^{n_{pu}} \tilde{\xi}_l & \\
\leq \tilde{\boldsymbol{\beta}}^T \tilde{\boldsymbol{\beta}} + \frac{c'}{n_{pu}} \left(\tau n_1^- + (2+\tau) n_2 + (1+\tau) n_3 + (2+\tau) n_4\right) & \\
= \tilde{\boldsymbol{\beta}}^T \tilde{\boldsymbol{\beta}} + \frac{c'}{n_{pu}} \left(\tau n_1^- + (2+\tau)(n_{pu} - n_1) - n_3\right). &
\end{aligned} \tag{5.74}$$

Then according to Inequality 5.64 and the condition in Inequality 5.65, the same as the content in [80], there is

$$\begin{aligned}
&(2+\tau)\,(n_{pu}-n_1)\\
\le& n_{pu}(2+\tau)\left(1-\sum_{j=-1,1}\sum_{i=0}^{2^{\rho-1}-2}\sum_{\mathscr{A}\in\mathbb{A}_i^j}(1-\tau)\left(1-\frac{i+1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{A}]\right)\\
\le& 2n_{pu}-2n_{pu}(1-\tau)\sum_{i=0}^{2^{\rho-1}-2}\left(1-\frac{i+1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}\cup\mathscr{B}_{[i]}^{-1}]+5n_{pu}\tau\\
=& 2n_{pu}(1-\tau)\left(1-\sum_{i=0}^{2^{\rho-1}-2}\left(1-\frac{i+1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}\cup\mathscr{B}_{[i]}^{-1}]\right)+7n_{pu}\tau\\
\le&\left(1-\sum_{i=0}^{2^{\rho-1}-2}P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}\cup\mathscr{B}_{[i]}^{-1}]+\sum_{i=0}^{2^{\rho-1}-2}\frac{i}{2^{\rho}}P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}\cup\mathscr{B}_{[i]}^{-1}]\right)2n_{pu}(1-\tau)+9n_{pu}\tau.
\end{aligned}\tag{5.75}$$

According to Inequality 5.60, Inequality 5.64 and Inequality 5.65, we have

$$\begin{aligned}
&\frac{\tau}{n_{pu}}n_n^- \le\tau\left[1-\sum_{i=0}^{2^{\rho-1}-2}\sum_{\mathscr{A}\in\mathbb{A}_i^1}(1-\tau)\left(1-\frac{i+1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{A}]\right]\\
&\le\tau\left[1-\frac{1}{2}(1-\tau)\sum_{i=0}^{2^{\rho-1}-2}\sum_{\mathscr{A}\in\mathbb{A}_i^1}P[\boldsymbol{X}\in\mathscr{A}]\right]\\
&\le\tau\left[1-\frac{1}{2}(1-\tau)\sum_{i=0}^{2^{\rho-1}-2}(P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}]-\tau^2)\right]\\
&=\tau\left\{1-\frac{1}{2}(1-\tau)\left[\sum_{i=0}^{2^{\rho-1}-2}P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}]-(2^{\rho-1}-1)\tau^2\right]\right\}\\
&\le\tau\left[1-\frac{1}{2}(1-\tau)\left(\sum_{i=0}^{2^{\rho-1}-2}P[\boldsymbol{X}\in\mathscr{B}_{[i]}^{1}]-\tau\right)\right]\\
&\le\tau\left[1-\frac{1}{2}(1-\tau)\,(D_- -2\tau)\right]
\end{aligned}\tag{5.76}$$

where $D_-=P[\boldsymbol{X}\in\boldsymbol{Z}_+]-P[\boldsymbol{X}\in\mathscr{S}_{x[2^{\rho-1}-1]}\cap\boldsymbol{Z}_+]$.

Besides, according to Inequality 5.64 and Inequality 5.66, the same as the content in [80],there is

$$\begin{aligned}
n_3 &\geq 2n_{pu}(1-\tau)\left[\sum_{\mathscr{A}\in\mathbb{A}^{1}_{2^{\rho-1}-1}}\left(\frac{1}{2}-\frac{1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{A}]+\sum_{\mathscr{A}\in\mathbb{A}^{-1}_{2^{\rho-1}-1}}\left(\frac{1}{2}-\frac{1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{A}]\right]\\
&\geq 2n_{pu}(1-\tau)\left(\frac{1}{2}-\frac{1}{2^{\rho}}\right)\left(P[\boldsymbol{X}\in\mathscr{B}_{[2^{\rho-1}-1]}]-2\tau\right)\\
&\geq 2n_{pu}(1-\tau)\left\{P[\boldsymbol{X}\in\mathscr{B}_{[2^{\rho-1}-1]}]-\left(\frac{1}{2}-\frac{1}{2^{\rho}}\right)P[\boldsymbol{X}\in\mathscr{B}_{[2^{\rho-1}-1]}]\right\}-6n_{pu}\tau.
\end{aligned}\tag{5.77}$$

Combining Inequality 5.75 and Inequality 5.77 with Inequality 5.59, Inequality 5.60, and Inequality 5.61, we can get

$$\begin{aligned}
&\frac{1}{n_{pu}}\left((2+\tau)\left(n_{pu}-n_1\right)-n_3\right)\\
&\leq 2(1-\tau)\left(1-\sum_{i=0}^{2^{\rho-1}-1}P[\boldsymbol{X}\in\mathscr{B}^{1}_{[i]}\cup\mathscr{B}^{-1}_{[i]}]+\sum_{i=0}^{2^{\rho-1}-1}\frac{i}{2^{\rho}}P[\boldsymbol{X}\in\mathscr{B}^{1}_{[i]}\cup\mathscr{B}^{-1}_{[i]}]\right)+15\tau\\
&\leq 2(1-\tau)\left(\tau+\sum_{i=0}^{2^{\rho-1}-1}\frac{i}{2^{\rho}}P\left[\boldsymbol{X}\in\mathscr{S}_{x[i]}\right]\right)+15\tau\\
&\leq 2(1-\tau)(\mathscr{R}_{\text{Bayes}}+\tau)+15\tau\\
&\leq 2(1-\tau)(\mathscr{R}_{\text{Bayes}}+8.75\tau).
\end{aligned}\tag{5.78}$$

Combining Inequality 5.75, Inequality 5.76 and Inequality 5.77, we can eventually obtain

$$\begin{aligned}\boldsymbol{\beta}_{[\mathrm{ap}]}^{T}\boldsymbol{\beta}_{[\mathrm{ap}]}+\frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l} &\leq \tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}+2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}}+(8.75+\frac{n_1^-}{2n_{pu}(1-\tau)})\tau\right]\\ &\leq \tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}+2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}}+(8.75+\frac{n_n}{2n_{pu}(1-\tau)})\tau\right]\\ &\leq \tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}+2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}}+(8.75+\frac{1-\frac{1}{2}(1-\tau)(D_{-}-2\tau)}{2(1-\tau)})\tau\right]\end{aligned} \tag{5.79}$$

5.8.6.2 Lower Bound of the Inequality for Contradiction

If we substitute $\boldsymbol{\beta}=\mathbf{0}$ and $\beta_0=0$ into the objective function in Equation 5.54, it is obvious to find

$$\boldsymbol{\beta}_{[\mathrm{ap}]}^{T}\boldsymbol{\beta}_{[\mathrm{ap}]}\leq \boldsymbol{\beta}_{[\mathrm{ap}]}^{T}\boldsymbol{\beta}_{[\mathrm{ap}]}+\frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l}\leq 0+\frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}1=c'. \tag{5.80}$$

Then we are to discuss the lower bound of $\sum\xi_{[\mathrm{ap}]l}$ with $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^j$ and $\mathscr{A}\cap E_i^j\neq\emptyset$ for $i=0,1,\ldots,2^{\rho-1}-2$. To simplify the analysis firstly we suppose $j=1$. Let $\boldsymbol{x}_z=\arg\min_{\boldsymbol{x}}\left(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)$ s.t., $\boldsymbol{x}\in\mathscr{A}\cap\mathscr{E}_i^1$. Then, we define $f_z=-\left(\boldsymbol{\phi}(\boldsymbol{x}_z)^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)\geq 0$. By the definition of $\boldsymbol{\sigma}$ in Section 5.8.5, for $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^1$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n_{pu}}\}$, we have $\|\boldsymbol{\phi}(\boldsymbol{x}_l)-\boldsymbol{\phi}(\boldsymbol{x}_z)\|_2=d_{\boldsymbol{\phi}}(\boldsymbol{x}_l,\boldsymbol{x}_z)\leq\sigma=\frac{\tau}{\sqrt{c'}}$. Supposing $y_l=1$, according to the Cauchy–Schwarz inequality and Inequality 5.80, there is

$$\begin{aligned}1-\xi_{[\mathrm{ap}]l} &\leq \boldsymbol{\phi}(x_l)^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\\ &=(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\boldsymbol{\phi}(\boldsymbol{x}_z)\boldsymbol{\beta}_{[\mathrm{ap}]}^T+\beta_{[\mathrm{ap}]0}\\ &=(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}-f_z\\ &\leq|(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}|-f_z\\ &\leq\|\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z)\|_2\cdot\left\|\boldsymbol{\beta}_{[\mathrm{ap}]}^T\right\|_2-f_z\\ &\leq\tau-f_z.\end{aligned} \tag{5.81}$$

Thus there is $\xi_{[\mathrm{ap}]l} \geq 1-\tau+f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^1 \neq \emptyset$ and $y_l = 1$.

Furthermore, suppose that there exists an $\boldsymbol{x}_k \in \mathscr{A} \in \mathbb{A}_i^1$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\left(\boldsymbol{\phi}(\boldsymbol{x}_k)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \leq -1$, which suffices to $\boldsymbol{x}_k \in \mathscr{A} \cup \mathscr{E}_i^1$ so that $\boldsymbol{f}_z \geq 1$ in this case. For $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $y_l = -1$ and $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \leq -1$, there is

$$\begin{aligned}
1-\xi_{[\mathrm{ap}]l} &\leq 1-|1+\boldsymbol{\beta}_{[\mathrm{ap}]}^T \boldsymbol{\phi}(\boldsymbol{x}_l)+\beta_{[\mathrm{ap}]0}| \\
&= 2+\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0} \\
&= 2+(\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}+\boldsymbol{\phi}(\boldsymbol{x}_z)\boldsymbol{\beta}_{[\mathrm{ap}]}^T+\beta_{[\mathrm{ap}]0} \\
&= (\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}-f_z \qquad (5.82)\\
&\leq 2+|(\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}|-f_z \\
&\leq 2+\|\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z)\|_2 \cdot \left\|\boldsymbol{\beta}_{[\mathrm{ap}]}^T\right\|_2-f_z \\
&\leq 2+\tau-f_z.
\end{aligned}$$

Then for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ with $y_l = -1$ and $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) > -1$, there is

$$\begin{aligned}
1-\xi_{[\mathrm{ap}]l} &\leq 1-|1+\boldsymbol{\beta}_{[\mathrm{ap}]}^T \boldsymbol{\phi}(\boldsymbol{x}_l)+\beta_{[\mathrm{ap}]0}| \\
&= -\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]}-\beta_{[\mathrm{ap}]0} \\
&= -(\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}-\boldsymbol{\phi}(\boldsymbol{x}_z)\boldsymbol{\beta}_{[\mathrm{ap}]}^T-\beta_{[\mathrm{ap}]0} \\
&= -(\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}+f_z \qquad (5.83)\\
&\leq |(\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}|+f_z \\
&\leq \|\boldsymbol{\phi}\,(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z)\|_2 \cdot \left\|\boldsymbol{\beta}_{[\mathrm{ap}]}^T\right\|_2+f_z \\
&\leq \tau+f_z.
\end{aligned}$$

Since $\boldsymbol{f}_z \geq 1$ here, we have $\tau+f_z \geq 2+\tau-f_z$. Therefore, once there exists an $\boldsymbol{x}_k \in \mathscr{A}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\left(\boldsymbol{\phi}(\boldsymbol{x}_k)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \leq -1$, we have $1 - \xi_{[\mathrm{ap}]l} \leq \tau + f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ with $y_l = -1$.

Moreover, suppose that $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) > -1$ holds for any $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ with $\mathscr{A} \cap \mathscr{E}_i^1 \neq \emptyset$. In this case we only need to consider Inequality 5.83, i.e., $1 - \xi_{[\mathrm{ap}]l} \leq \tau + f_z$ for $y_l = -1$. Thus, generally we have $\xi_{[\mathrm{ap}]l} \geq 1 - \tau - f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^1 \neq \emptyset$ and $y_l = -1$. Then according to Inequality 5.65, we get the same result as the content in [80]:

$$\begin{aligned}
\frac{1}{n_{pu}} \sum_{x_l \in \mathscr{A}} \xi_{[\mathrm{ap}]l} \geq\ & (1 - \tau + f_z)(1 - \tau)\left(1 - \frac{i+1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\
& + (1 - \tau - f_z)(1 - \tau)\frac{i}{2^\rho} P[\boldsymbol{X} \in \mathscr{A}] \\
& = (1 - \tau)^2 \left(1 - \frac{1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\
& + f_z(1 - \tau)\left(1 - \frac{2i+1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\
& \geq (1 - \tau)^2 \left(1 - \frac{1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}].
\end{aligned} \tag{5.84}$$

For $\mathscr{A} \in \mathbb{A}_i^{-1}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} \neq \emptyset$, let $\boldsymbol{x}_z = \arg\max_{\boldsymbol{x}} \left(\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right)$ s.t., $\boldsymbol{x} \in \mathscr{A} \cap E_i^{-1}$. Then, we define $f_z = \left(\boldsymbol{\phi}(\boldsymbol{x}_z)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \geq 0$. For $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $y_l = 1$, there is

$$\begin{aligned}
1 - \xi_{[\mathrm{ap}]l} & \leq \boldsymbol{\phi}(x_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0} \\
& = (\boldsymbol{\phi}(x_l) - \boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \boldsymbol{\phi}(\boldsymbol{x}_z)\boldsymbol{\beta}_{[\mathrm{ap}]}^T + \beta_{[\mathrm{ap}]0} \\
& = (\boldsymbol{\phi}(x_l) - \boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]} + f_z \\
& \leq |(\boldsymbol{\phi}(x_l) - \boldsymbol{\phi}(\boldsymbol{x}_z))^T \boldsymbol{\beta}_{[\mathrm{ap}]}| + f_z \\
& \leq \|\boldsymbol{\phi}(x_l) - \boldsymbol{\phi}(\boldsymbol{x}_z)\|_2 \cdot \left\|\boldsymbol{\beta}_{[\mathrm{ap}]}^T\right\|_2 + f_z \\
& \leq \tau + f_z,
\end{aligned} \tag{5.85}$$

so that there is $\xi_{[\mathrm{ap}]l} \geq 1 - \tau - f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} \neq \emptyset$ and $y_l = 1$. Moreover, $\mathscr{A} \cap \mathscr{E}_i^{-1} \neq \emptyset$ indicates that there exists $\boldsymbol{x} \in \mathscr{A} \cap \mathscr{E}_i^{-1}$ meeting

$\left(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)>0$. Considering the continuity of $\boldsymbol{\phi}(\cdot)$ and the diameter of $\mathscr{A}\in\tilde{\mathbb{A}}_i^{-1}$ no greater than σ as discussed in Section 5.8.5, $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)\geq -1$ holds for any $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n_{pu}}\}$ when there exists $\boldsymbol{x}\in\mathscr{A}$ meeting $\left(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)>0$. In this case, for $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^{-1}$ with $y_l=-1$, we have

$$\begin{aligned}1-\xi_{[\mathrm{ap}]l}&\leq-\boldsymbol{\phi}(\boldsymbol{x}_l)^T\boldsymbol{\beta}_{[\mathrm{ap}]}-\beta_{[\mathrm{ap}]0}\\&=-(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}-\boldsymbol{\phi}(\boldsymbol{x}_z)\boldsymbol{\beta}_{[\mathrm{ap}]}^T-\beta_{[\mathrm{ap}]0}\\&=-(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}-f_z\\&\leq|(\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z))^T\boldsymbol{\beta}_{[\mathrm{ap}]}|-f_z\\&\leq\|\boldsymbol{\phi}(x_l)-\boldsymbol{\phi}(\boldsymbol{x}_z)\|_2\cdot\left\|\boldsymbol{\beta}_{[\mathrm{ap}]}^T\right\|_2-f_z\\&\leq\tau-f_z,\end{aligned}\tag{5.86}$$

so that there is $\xi_{[\mathrm{ap}]l}\geq 1-\tau+f_z$ for $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A}\cap\mathscr{E}_i^{-1}\neq\emptyset$ and $y_l=-1$. In this case, we can obtain the same result as Inequality 5.84, i.e., $\frac{1}{n_{pu}}\sum_{x_l\in\mathscr{A}}\xi_{[\mathrm{ap}]l}\geq(1-\tau)^2\left(1-\frac{1}{2^\rho}\right)P[\boldsymbol{X}\in\mathscr{A}]$.

Thus, according to Inequality 5.84, we can obtain the same result as the content in [80]:

$$\frac{1}{n_{pu}}\sum_{j=-1,1}\sum_{\substack{\mathscr{A}\in\mathbb{A}_i^j\\\mathscr{A}\cap\mathscr{E}_i^j\neq\emptyset}}\sum_{\boldsymbol{x}_l\in\mathscr{A}}\xi_{[\mathrm{ap}]l}\geq(1-\tau)^2\left(1-\frac{1}{2^\rho}\right)\sum_{\substack{\mathscr{A}\in\mathbb{A}_i^j\\\mathscr{A}\cap\mathscr{E}_i^j\neq 0}}P[\boldsymbol{X}\in\mathscr{A}].\tag{5.87}$$

Then we are to discuss the lower bound of $\sum\xi_{[\mathrm{ap}]l}$ with $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^j$ and $\mathscr{A}\cap\mathscr{E}_i^j=\emptyset$ for $i=0,1,\ldots,2^{\rho-1}-2$. Firstly we suppose $j=1$. Let $\boldsymbol{x}_z=\arg\max_{\boldsymbol{x}}\left(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)$ s.t., $\boldsymbol{x}\in\mathscr{A}$. Then define $f_z=\left(\boldsymbol{\phi}(\boldsymbol{x}_z)^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0}\right)\geq 0$. For $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^1$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=1$, there is the same analysis as Inequality 5.85, i.e.,$1-\xi_{[\mathrm{ap}]l}\leq\tau+f_z$ so that $\xi_{[\mathrm{ap}]l}\geq\max 1-\tau-a$.

Furthermore, $\mathscr{A}\cap\mathscr{E}_i^1=\emptyset$ indicates that $(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}_{[\mathrm{ap}]}+\beta_{[\mathrm{ap}]0})>0$ holds for any $\boldsymbol{x}\in\mathscr{A}$. Thus, for $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^1$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=-1$, we only need to consider the case in Inequality 5.86, i.e., $1-\xi_{[\mathrm{ap}]l}\leq\tau-f_z$ so that $\xi_{[\mathrm{ap}]l}\geq 1-\tau+f_z$

for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^1 = \emptyset$ and $y_l = -1$. According to Inequality 5.65, we can get the same result as the content in [80], i.e.,

$$\begin{aligned}\frac{1}{n_{pu}} \sum_{\boldsymbol{x}_l \in \mathscr{A}} \xi_{[\mathrm{ap}]l} \geq &(1-\tau-f_z)(1-\tau)\left(1-\frac{i+1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\ &+ (1-\tau+f_z)(1-\tau)\frac{i}{2^\rho} P[\boldsymbol{X} \in \mathscr{A}] \\ &= (1-\tau)^2\left(1-\frac{1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\ &- f_z(1-\tau)\left(1-\frac{2i+1}{2^\rho}\right) P[\boldsymbol{X} \in \mathscr{A}] \\ &\geq (1-\tau)^2 \frac{i}{2^{\rho-1}} P[\boldsymbol{X} \in \mathscr{A}].\end{aligned} \tag{5.88}$$

Then we are to discuss the lower bound of $\sum \xi_{[\mathrm{ap}]l}$ with $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ and $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$ for $i = 0, 1, \ldots, 2^{\rho-1}-2$. Let $\boldsymbol{x}_z = \arg\min_{\boldsymbol{x}} \left(\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right)$ s.t., $\boldsymbol{x} \in \mathscr{A}$. Then define $f_z = -\left(\boldsymbol{\phi}(\boldsymbol{x}_z)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \geq 0$. For $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^1$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$ and $y_l = 1$, there is the same result as Inequality 5.81, i.e.,$1-\xi_{[\mathrm{ap}]l} \leq \tau - f_z$ so that $\xi_{[\mathrm{ap}]l} \geq 1-\tau+f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$ and $y_l = 1$.

Furthermore, suppose that there exists an $\boldsymbol{x}_k \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\left(\boldsymbol{\phi}(\boldsymbol{x}_k)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \leq -1$, which suffices to $\boldsymbol{f}_z \geq 1$ in this case. For $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $y_l = -1$ and $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) \leq -1$, there is the same result as the Inequality 5.82, i.e., $\xi_{[\mathrm{ap}]l} \leq 2+\tau-f_z$. For $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $y_l = -1$ and $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) > -1$, there is the same result as the Inequality 5.83, i.e., $\xi_{[\mathrm{ap}]l} \leq \tau + f_z$.

Moreover, suppose that $\left(\boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{\beta}_{[\mathrm{ap}]} + \beta_{[\mathrm{ap}]0}\right) > -1$ holds for any $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$. In this case we only need to consider Inequality 5.83, i.e., $1-\xi_{[\mathrm{ap}]l} \leq \tau + f_z$ for $y_l = -1$. Thus, generally we have $\xi_{[\mathrm{ap}]l} \geq 1-\tau-f_z$ for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{pu}}\}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$ and $y_l = -1$. Therefore, Inequality 5.88 also holds for $\boldsymbol{x}_l \in \mathscr{A} \in \mathbb{A}_i^{-1}$ with $\mathscr{A} \cap \mathscr{E}_i^{-1} = \emptyset$ so that there is the same result as in [80]:

$$\frac{1}{n_{pu}}\sum_{j=-1,1}\sum_{\substack{\mathscr{A}\in\mathbb{A}_i^j\\ \mathscr{A}\cap E_i^j=0}}\sum_{\boldsymbol{x}_l\in\mathscr{A}}\xi_{[\text{ap}]l}\geq(1-\tau)^2\frac{i}{2^{\rho-1}}\sum_{\substack{\mathscr{A}\in\mathbb{A}_i^j\\ \mathscr{A}\cap E_i^j=0}}P[\boldsymbol{X}\in\mathscr{A}]. \tag{5.89}$$

Finally, as for the lower bound of $\sum\xi_{[\text{ap}]l}$ with $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}^j_{2^{\rho-1}-1}$, the analysis is very similar to the above analysis for the lower bound of $\sum\xi_{[\text{ap}]l}$ with $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}_i^j$ and $\mathscr{A}\cap\mathscr{E}_i^j=\emptyset$ for $i=0,1,\dots,2^{\rho-1}-2$. There for we skip to the results as:

- For $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}^1_{2^{\rho-1}-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=1$, there is $1-\xi_{[\text{ap}]l}\leq\tau+f_z$ so that $\xi_{[\text{ap}]l}\geq1-\tau-a$
- For $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}^1_{2^{\rho-1}-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=-1$, there is $1-\xi_{[\text{ap}]l}\leq\tau-f_z$ so that $\xi_{[\text{ap}]l}\geq1-\tau+f_z$
- For $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}^{-1}_{2^{\rho-1}-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=1$, there is $1-\xi_{[\text{ap}]l}\leq\tau-f_z$ so that $\xi_{[\text{ap}]l}\geq1-\tau+f_z$
- For $\boldsymbol{x}_l\in\mathscr{A}\in\mathbb{A}^{-1}_{2^{\rho-1}-1}$ from $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_{n_{pu}}\}$ with $y_l=-1$, there is $1-\xi_{[\text{ap}]l}\leq\tau+f_z$ so that $\xi_{[\text{ap}]l}\geq1-\tau-a$

According to Inequality 5.64 and Inequality, there is the same result as the content in [80] with , i.e.,

$$\begin{aligned}\frac{1}{n_{pu}}\sum_{j=-1,1}\sum_{\mathscr{A}\in\mathbb{A}^j_{2^{\rho-1}-1}}\sum_{\boldsymbol{x}_l\in\mathscr{A}}\xi_{[\text{ap}]l}\geq P[\boldsymbol{X}\in\mathscr{B}^{*1}_{2^{\rho-1}-1}\cup\mathscr{B}^{*-1}_{2^{\rho-1}-1}]&\left[(1-\tau-f_z)(1-\tau)\left(\frac{1}{2}-\frac{1}{2^\rho}\right)\right.\\ &\left.+(1-\tau+f_z)(1-\tau)\left(\frac{1}{2}-\frac{1}{2^\rho}\right)\right]\\ &>(1-\tau)^2\left(1-\frac{1}{2^{\rho-1}}\right)\left(P[\boldsymbol{X}\in\mathscr{B}_{2^{\rho-1}-1}]-2\tau\right).\end{aligned} \tag{5.90}$$

Up to now, we have got the three same results as [80] in Inequality 5.87, Inequality 5.89 and Inequality 5.90. We can eventually obtain the following lower bound for $\sum_{l=1}^{n_{pu}}\xi_{[\text{ap}]l}$ by summing Inequality 5.87, Inequality 5.89 and Inequality

5.90 together, which has been proved in [80].

$$\frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l} \geq (1-\tau)^2 c' \left(2\mathscr{R}_{\mathrm{Bayes}} + \sum_{j=-1,1}\sum_{i=0}^{2^{\rho-1}-2}\left(1-\frac{i}{2^{\rho-1}}\right)P[\boldsymbol{X}\in\mathscr{E}_i^j] - 9\tau\right). \tag{5.91}$$

Considering $\mathscr{R}_{\mathrm{Bayes}} \leq \frac{1}{2}$ and Inequality 5.71, there is

$$\begin{aligned}\frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l} &> (1-\tau)^2 c' \left(2\mathscr{R}_{\mathrm{Bayes}} + \varepsilon - 11\tau\right)\\ &= c'(1-\tau)\left(2\mathscr{R}_{\mathrm{Bayes}} + 32\tau - 11\tau - 2\tau\mathscr{R}_{\mathrm{Bayes}} - \varepsilon\tau + 11\tau^2\right)\\ &> c'(1-\tau)\left(2\mathscr{R}_{\mathrm{Bayes}} + 19\tau\right).\end{aligned} \tag{5.92}$$

5.8.6.3 Construction of Contradiction for the Proof of Theorem 4

Combining Inequality 5.79 with Inequality 5.92 we can find

$$\begin{aligned}\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}} &\geq \boldsymbol{\beta}_{[\mathrm{ap}]}^T\boldsymbol{\beta}_{[\mathrm{ap}]} + \frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l}\\ &\quad - 2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}} + (8.75 + \frac{1-\frac{1}{2}(1-\tau)(D_- - 2\tau)}{2(1-\tau)})\tau\right]\\ &\geq \frac{c'}{n_{pu}}\sum_{l=1}^{n_{pu}}\xi_{[\mathrm{ap}]l} - 2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}} + (8.75 + \frac{1-\frac{1}{2}(1-\tau)(D_- - 2\tau)}{2(1-\tau)})\tau\right]\\ &> c'(1-\tau)\left(2\mathscr{R}_{\mathrm{Bayes}} + 19\tau\right)\\ &\quad - 2c'(1-\tau)\left[\mathscr{R}_{\mathrm{Bayes}} + (8.75 + \frac{1-\frac{1}{2}(1-\tau)(D_- - 2\tau)}{2(1-\tau)})\tau\right]\\ &= c'\tau\left[0.5 - 1.5\tau + \frac{1}{2}(1-\tau)(D_- - 2\tau)\right].\end{aligned} \tag{5.93}$$

It should be noted that

$$\tau \leq \frac{1}{32} < 0.2 < \frac{1}{4}\min_{D_-\in[0,1]}\{5 + D_- - \sqrt{(5+D_-)^2 - 8(D_- + 1)}\}. \tag{5.94}$$

Therefore, $0.5-1.5\tau+\frac{1}{2}(1-\tau)(D_{-}-2\tau)>0$ holds for all $0<\tau=\frac{\varepsilon}{32}\leq\frac{1}{32}$. Then, we let

$$\begin{aligned} c^{*} &= \frac{\tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}}{\tau\min_{D_{-}\in[0,1]}\{0.5-1.5\tau+\frac{1}{2}(1-\tau)(D_{-}-2\tau)\}} \\ &= \frac{2\tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}}{\tau(1-5\tau+2\tau^{2})}. \end{aligned} \tag{5.95}$$

For $c'\geq c^{*}$, we can finally obtain the contradiction according to Inequality 5.93, i.e.,

$$\tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}>c^{*}\tau\left[0.5-1.5\tau+\frac{1}{2}(1-\tau)(D_{-}-2\tau)\right]>\tilde{\boldsymbol{\beta}}^{T}\tilde{\boldsymbol{\beta}}. \tag{5.96}$$

Thus the assumption in Inequality 5.72 is false and we can draw a conclusion that for $0<\varepsilon=32\tau<1$ and $c'\geq c^{*}$,

$$\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)-\mathscr{R}_{\text{Bayes}}\leq\varepsilon$$

holds on the 'representative' dataset. Finally Theorem 4 is proved.

5.8.7 Step 3: Proof of Theorem 3

As the value of c' tends to infinity with n_{pu} increasing, we can find n^{*} so that $c'\geq c^{*}$ when there is $n_{pu}\geq n^{*}$. In this case, according to Theorem 4, there is

$$P^{n_{pu}}\left[\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)-\mathscr{R}_{\text{Bayes}}\leq\varepsilon\right]\geq 1-2M_{n_{pu}}e^{-\left(\varepsilon^{6}/2^{29}M_{n_{pu}}^{2}\right)n_{pu}}, \tag{5.97}$$

where $M_{n_{pu}}=\frac{64}{\varepsilon}\mathscr{N}\left(\left(\mathscr{S}_{x},d_{\boldsymbol{\phi}}\right),\frac{\varepsilon}{32\sqrt{c'}}\right)$. There is $M_{n_{pu}}^{2}\in\mathscr{O}\left(c'^{\alpha}\right)$ according to the assumption on the covering numbers of kernel space $(\mathscr{S}_{x},d_{\boldsymbol{\phi}})$. Therefore, $n_{pu}M_{n_{pu}}^{-2}$ tends to infinity with n_{pu} increasing and there is

$$P^{n_{pu}}\left[\mathscr{R}_{0-1}\left(f_{\text{ap}}^{*n_{pu}}\right)-\mathscr{R}_{\text{Bayes}}\leq\varepsilon\right]\rightarrow 1.$$

As $\mathscr{R}_{0-1}\left(f^{*n_{pu}}_{\text{cpb}}\right) \rightarrow \mathscr{R}_{0-1}\left(f^{*n_{pu}}_{\text{ap}}\right)$ with n_{pu} increasing, Theorem 3 is proved, i.e.,

$$P^{n_{pu}}\left[\mathscr{R}_{0-1}\left(f^{*n_{pu}}_{\text{cpb}}\right) - \mathscr{R}_{\text{Bayes}} \leq \varepsilon\right] \rightarrow 1.$$

5.8.8 Discussion on Universal Consistency of CPB-GLPUAL.

There is still one problem left. The construction of the kernel function various according to the form of the objective function for optimisation. Currently, to have an algorithm to solve the optimisation, we did the kernel trick based on the blocked form of the objective function of CPB-GLPUAL in Section 5.6, where we can only determine the form of kernel matrices $\mathbf{\Phi}(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]}) = \boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{B}^{-1}\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T, k = p, u, pu$ as a whole directly. In this case, the form of kernel matrices $\mathbf{\Phi}^*(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]}) = \boldsymbol{\phi}(\boldsymbol{X}_{[k]})\boldsymbol{\phi}(\boldsymbol{X}_{[pu]})^T, k = p, u, pu$ is unknown due to the lack of expertise. As a compromise, during the experiments on real datasets in Section 5.7, we set $\mathbf{\Phi}(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$ to be universal kernel matrices, i.e., RBF kernel matrices, to intuitively increase the likelihood of $\mathbf{\Phi}^*(\boldsymbol{X}_{[k]}, \boldsymbol{X}_{[pu]})$ to be universal matrices.

Therefore, in future, we need to construct the kernel trick of CPB-GLPUAL in a way more consistent to the construction of kernel in Section 5.8.3.1 and then to propose the algorithm for the corresponding the optimisation. In this case, we will be able to completely ensure that the universal consistency holds in practice as universal kernel is applied.

5.9 Conclusions

In this chapter, firstly CPB-GLPUAL was proposed for better classification than GLPUAL with the class prior π known. Secondly, the algorithm to solve the non-convex optimisation of GLPUAL was proposed based on ADMM with the linear decision boundary in the original feature space generated. Thirdly, the kernel trick was introduced to GLPUAL and then the algorithm to solve the non-convex optimisation of GLPUAL was proposed also based on ADMM with the non-linear decision boundary generated in the original feature space. Fourthly, the motivation of CPB-GLPUAL was verified by the experiments on both synthetic and real datasets.

At the end of this chapter, the universal consistency of CPB-GLPUAL was proved to theoretically.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, three classifiers were proposed for better classification on various type of PU dataset.

In Chapter 3, firstly GLPUAL was proposed for better classification on the datasets where the distances from two positive subsets to the ideal decision boundary are very different. Secondly, an algorithm to solve the optimisation of GLPUAL was proposed based on ADMM with linear decision boundary generated. Thirdly, the kernel trick was introduced to GLPUAL and then the algorithm to solve the non-convex optimisation of GLPUAL was proposed also based on ADMM with the non-linear decision boundary generated in the original feature space for satisfactory classification on trifurcated PU datasets.

As SVM-based methods, both GLLC and GLPUAL are negatively affected by the irrelevant features in the datasets, especially when the kernel trick is applied [54]. This motivated us to introduce the elastic net [48] to the objective function of GLPUAL and the kernel-free technique [81] to the predictive score function of GLPUAL to propose E-GLPUAL and EKF-GLPUAL for the better performance than GLPUAL on the PU datasets with irrelevant features in Chapter 4.

In Chapter 4, we proposed E-GLPUAL and EKF-GLPUAL for better classification than GLPUAL on the datasets with irrelevant features contained. Then the algorithms to solve the optimisation for E-GLPUAL and EKF-GLPUAL were proposed based on ADMM. The experimental results on the synthetic datasets and real datasets support our motivation though currently EKF-GLPUAL cannot abandon all the irrelevant features thoroughly. In the end of Chapter 4, the grouping effect of E-GLPUAL is proved.

There are too many hyper-parameters to be tuned in the objective of GLPUAL, which may make the model miss the best combination of the hyper-parameters. Motivated by this issue, CPB-GLPUAL was proposed in Chapter 5 to reduce the number of the hyper-parameters to be tuned via introducing prior information of the datasets to the objective function.

In Chapter 5, firstly CPB-GLPUAL was proposed for better classification than GLPUAL with the class prior π known. Secondly, the algorithm to solve the non-convex optimisation of CPB-GLPUAL was proposed based on ADMM with the linear decision boundary in the original feature space generated. Thirdly, the kernel trick was introduced to CPB-GLPUAL and then the algorithm to solve the non-convex optimisation of GLPUAL was proposed also based on ADMM with the non-linear decision boundary generated in the original feature space. Fourthly, the universal consistency of CPB-GLPUAL was proved to theoretically ensure that extending the sample size can enhance the performance of CPB-GLPUAL.

6.2 Limitations and Implications

In Chapter 3, there is no closed form of the solution of the optimisation of GLPUAL and the numerical solution is to be obtained by the iterative algorithm based on ADMM, while the closed form of the optimisation of GLLC can be easily obtained. In this case, it takes much longer time to train GLPUAL than GLLC. More specifically, during the experiments in Chapter 3, we found that the training time for GLPUAL is approximately ten times longer than the training time for GLLC. After considering CV, the difference in training efficiency becomes even more distinct.

In Chapter 4, E-GLPUAL introduced a new hyper-parameter for the L1-norm regularisation term, making hyper-parameter tuning more complex compared to GLPUAL. This exacerbates the time consumption. The same holds true for EKF-

GLPUAL. Furthermore, before training EKF-GLPUAL, we have to convert each instance vector into a matrix form; this further extends the training time. Moreover, as mentioned in Section 4.5.1, computationally singular errors might encounter during the training of EKF-GLPUAL.

In Chapter 5, although there is one important hyper-parameter determined for CPB-GLPUAL, the non-convex objective function requires more iterative steps for the algorithm to converge to the optimal solution. Consequently, we still require more than seven times the training time for CPB-GLPUAL compared with GLLC.

Considering the above discussed limitations of the proposed methods in this thesis, it is time-costly to train all the methods mentioned in this thesis on a new PU dataset. In this case, for the convenience in selecting the method mentioned in this paper, the flowchart in Figure 6.1 is given. More specifically, the 2-dimensional projection with t-SNE is a strong tool to check if the PU data is trifurcated. Furthermore, if there are many features with low contribution during the principal component analysis (PCA), it will be reasonable to regard the PU data as the one with too many irrelevant features.

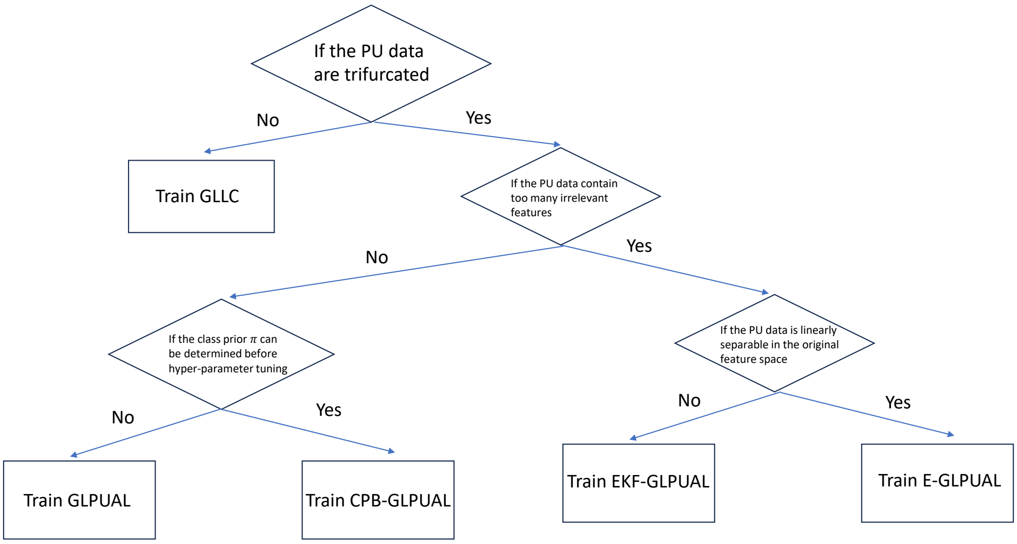
If the PU data are trifurcated
No
Yes
Train GLLC
If the PU data contain too many irrelevant features
No
Yes
If the class prior π can be determined before hyper-parameter tuning
If the PU data is linearly separable in the original feature space
No
Yes
No
Yes
Train GLPUAL
Train CPB-GLPUAL
Train EKF-GLPUAL
Train E-GLPUAL

Figure 6.1: Flowchart to determine which method mentioned in this thesis to use.

6.3 Future Work

Firstly, as discussed in Section 4.5.5, EKF-GLPUAL cannot abandon all the irrelevant features in the real datasets thoroughly. In this case, the main future work is to find a regularised term for the parameters, which is better on the feature selection under the framework of kernel-free SVMs.

Secondly, in Section 4.6, the grouping effect of EKF-GLPUAL is more complex to be explored than the grouping effect of E-GLPUAL. Therefore, the case-dependent studies on the grouping effect of EKF-GLPUAL is also regarded as the future work on EKF-GLPUAL.

Thirdly, as discussed in Section 5.8.8, we need to construct the kernel trick of CPB-GLPUAL in a way more consistent to the construction of kernel in Section 5.8.3.1 and then to propose the algorithm for the corresponding the optimisation. In this case, we will be able to completely ensure that the universal consistency holds in practice as universal kernel is applied.

Bibliography

[1] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Mach. Learn.*, 109(4):719–760, 2020.

[2] Yafeng Ren, Donghong Ji, and Hongbin Zhang. Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 488–498, 2014.

[3] Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. Spotting fake reviews via collective positive-unlabeled learning. In *2014 IEEE international conference on data mining*, pages 899–904. IEEE, 2014.

[4] Wenkai Li, Qinghua Guo, and Charles Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE transactions on geoscience and remote sensing*, 49(2):717–725, 2010.

[5] Songmin Dai, Xiaoqiang Li, Yue Zhou, Xichen Ye, and Tong Liu. GradPU: positive-unlabeled learning via gradient penalty and positive upweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7296–7303, 2023.

[6] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394. Sydney, NSW, 2002.

[7] Hwanjo Yu, Jiawei Han, and KC-C Chang. PEBL: Web page classification

without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

[8] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pages 587–592. Citeseer, 2003.

[9] Fengxiang He, Tongliang Liu, Geoffrey I Webb, and Dacheng Tao. Instance-dependent PU learning by Bayesian optimal relabeling. *arXiv preprint arXiv:1808.02180*, 2018.

[10] Dino Ienco, Ruggero G Pensa, and Rosa Meo. From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–25, 2012.

[11] Lu Liu and Tao Peng. Clustering-based Method for Positive and Unlabeled Text Categorization Enhanced by Improved TFIDF. *Journal of Information Science & Engineering*, 30(5), 2014.

[12] Sneha Chaudhari and Shirish Shevade. Learning from positive and unlabelled examples using maximum margin clustering. In *International Conference on Neural Information Processing*, pages 465–473. Springer, 2012.

[13] Ting Ke, Bing Yang, Ling Zhen, Junyan Tan, Yi Li, and Ling Jing. Building high-performance classifiers using positive and unlabeled examples for text classification. In *International symposium on neural networks*, pages 187–195. Springer, 2012.

[14] Yulin He, Xu Li, Manjing Zhang, Philippe Fournier-Viger, Joshua Zhexue Huang, and Salman Salloum. A novel observation points-based positive-unlabeled learning algorithm. *CAAI Transactions on Intelligence Technology*, 2023.

[15] Chengming Xu, Chen Liu, Siqian Yang, Yabiao Wang, Shijie Zhang, Lijie Jia, and Yanwei Fu. Split-PU: Hardness-aware Training Strategy for Positive-

Unlabeled Learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2719–2729, 2022.

[16] Emilio Dorigatti, Jann Goschenhofer, Benjamin Schubert, Mina Rezaei, and Bernd Bischl. Positive-Unlabeled Learning with Uncertainty-aware Pseudo-label Selection. *arXiv preprint arXiv:2201.13192*, 2022.

[17] Qianqiao Liang, Mengying Zhu, Yan Wang, Xiuyuan Wang, Wanjia Zhao, Mengyuan Yang, Hua Wei, Bing Han, and Xiaolin Zheng. Positive distribution pollution: rethinking positive unlabeled learning from a unified perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8737–8745, 2023.

[18] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *Third IEEE International Conference on Data Mining*, pages 179–186. IEEE, 2003.

[19] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[20] Zhigang Liu, Wenzhong Shi, Deren Li, and Qianqing Qin. Partially supervised classification: based on weighted unlabeled samples support vector machine. In *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, pages 1216–1230. IGI Global, 2008.

[21] Ting Ke, Ling Jing, Hui Lv, Lidong Zhang, and Yaping Hu. Global and local learning from positive and unlabeled examples. *Applied Intelligence*, 48(8):2373–2392, 2018.

[22] Chen Gong, Tongliang Liu, Jian Yang, and Dacheng Tao. Large-margin label-calibrated support vector machines for positive and unlabeled learning. *IEEE transactions on neural networks and learning systems*, 30(11):3471–3483, 2019.

[23] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27, 2014.

[24] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *arXiv preprint arXiv:1703.00593*, 2017.

[25] Guangxin Su, Weitong Chen, and Miao Xu. Positive-unlabeled learning from imbalanced data. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, Virtual Event*, 2021.

[26] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. A Variational Approach for Learning from Positive and Unlabeled Data. *Advances in Neural Information Processing Systems*, 33:14844–14854, 2020.

[27] Hengwei Zhao, Xinyu Wang, Jingtao Li, and Yanfei Zhong. Class Prior-Free Positive-Unlabeled Learning with Taylor Variational Loss for Hyperspectral Remote Sensing Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16827–16836, 2023.

[28] Yinjie Liu, Jie Zhao, and Yitian Xu. Robust and unbiased positive and unlabeled learning. *Knowledge-Based Systems*, 277:110819, 2023.

[29] Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*, 2018.

[30] Chen Gong, Qizhou Wang, Tongliang Liu, Bo Han, Jane J You, Jian Yang, and Dacheng Tao. Instance-Dependent Positive and Unlabeled Learning with Labeling Bias Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[31] Ting Ke, Hui Lv, Mingjing Sun, and Lidong Zhang. A biased least squares sup-

port vector machine based on Mahalanobis distance for PU learning. *Physica A: Statistical Mechanics and its Applications*, 509:422–438, 2018.

[32] Clayton Scott and Gilles Blanchard. Novelty detection: Unlabeled data definitely help. In *Artificial intelligence and statistics*, pages 464–471. PMLR, 2009.

[33] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[34] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] LVD Maaten. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579, 2008.

[36] Rajen Bhatt. Wireless Indoor Localization. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C51880.

[37] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976.

[38] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[39] Gui-Bo Ye and Xiaohui Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569, 2011.

[40] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pages 448–455, 2003.

[41] Kristen Jaskie and Andreas Spanias. Evaluating the Positive Unlabeled Learning Problem. In *Positive Unlabeled Learning*, pages 35–46. Springer, 2022.

[42] Shantanu Jain, Martha White, Michael W Trosset, and Predrag Radivojac. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944*, 2016.

[43] Marthinus Christoffel, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In *Asian Conference on Machine Learning*, pages 221–236. PMLR, 2016.

[44] Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[45] Marc G Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, 2(Dec):299–312, 2001.

[46] Jonas Ranstam and JA Cook. LASSO regression. *Journal of British Surgery*, 105(10):1348–1348, 2018.

[47] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for SVMs. *Advances in neural information processing systems*, 13, 2000.

[48] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[49] Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, pages 589–615, 2006.

[50] Quan Zhou, Wenlin Chen, Shiji Song, Jacob Gardner, Kilian Weinberger, and Yixin Chen. A reduction of the elastic net to support vector machines with an application to GPU computing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[51] Amin Ul Haq, Jian Ping Li, Muhammad Hammad Memon, Asad Malik, Tanvir Ahmad, Amjad Ali, Shah Nazir, Ijaz Ahad, Mohammad Shahid, et al. Feature

selection based on L1-norm support vector machine and effective recognition system for Parkinson's disease using voice recordings. *IEEE access*, 7:37718–37734, 2019.

[52] Min Tan, Gang Pan, Yueming Wang, Yuting Zhang, and Zhaohui Wu. L1-norm latent SVM for compact features in object detection. *Neurocomputing*, 139:56–64, 2014.

[53] Ahmad Mousavi, Zheming Gao, Lanshan Han, and Alvin Lim. Quadratic surface support vector machine with L1 norm regularization. *arXiv preprint arXiv:1908.08616*, 2019.

[54] Hai Thanh Nguyen, Katrin Franke, and Slobodan Petrovi'c. On general definition of L1-norm support vector machines for feature selection. *International Journal of Machine Learning and Computing*, 1(3):279, 2011.

[55] Jian Luo, Shu-Cherng Fang, Zhibin Deng, and Xiaoling Guo. Soft quadratic surface support vector machine for binary classification. *Asia-Pacific Journal of Operational Research*, 33(06):1650046, 2016.

[56] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

[57] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1):57–79, 2016.

[58] Kun Dai, Hong-Yi Yu, and Qing Li. A semisupervised feature selection with support vector machine. *Journal of Applied Mathematics*, 2013, 2013.

[59] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(1):53–71, 2008.

[60] Daniel Tabak and Benjamin C Kuo. *Optimal control by mathematical programming*. SRL Publishing Company, 1971.

[61] Gill Ward, Trevor Hastie, Simon Barry, Jane Elith, and John R Leathwick. Presence-only data and the EM algorithm. *Biometrics*, 65(2):554–563, 2009.

[62] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

[63] Saurabh Garg, Yifan Wu, Alexander J Smola, Sivaraman Balakrishnan, and Zachary Lipton. Mixture Proportion Estimation and PU Learning: A Modern Approach. *Advances in Neural Information Processing Systems*, 34, 2021.

[64] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Gang Niu, Masashi Sugiyama, and Dacheng Tao. Rethinking class-prior estimation for positive-unlabeled learning. *arXiv preprint arXiv:2002.03673*, 2020.

[65] Harish Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *International conference on machine learning*, pages 2052–2060. PMLR, 2016.

[66] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[67] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*, pages 1386–1394. PMLR, 2015.

[68] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78:29–63, 2019.

[69] Mingyi Hong. A distributed, asynchronous and incremental algorithm for nonconvex optimization: An admm based approach. *arXiv preprint arXiv:1412.6058*, 2014.

[70] Xiaoquan Wang, Hu Shao, Pengjie Liu, and Ting Wu. An inertial proximal partially symmetric ADMM-based algorithm for linearly constrained multi-block nonconvex optimization problems with applications. *Journal of Computational and Applied Mathematics*, 420:114821, 2023.

[71] Maryam Yashtini. Multi-block Nonconvex Nonsmooth Proximal ADMM: Convergence and Rates Under Kurdyka–Łojasiewicz Property. *Journal of Optimization Theory and Applications*, 190(3):966–998, 2021.

[72] Maryam Yashtini. Convergence and rate analysis of a proximal linearized ADMM for nonconvex nonsmooth optimization. *Journal of Global Optimization*, 84(4):913–939, 2022.

[73] Qinghua Liu, Xinyue Shen, and Yuantao Gu. Linearized ADMM for nonconvex nonsmooth optimization with convergence analysis. *IEEE access*, 7:76131–76144, 2019.

[74] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[75] Masashi Sugiyama. *Introduction to statistical machine learning*. Morgan Kaufmann, 2015.

[76] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE transactions on information theory*, 51(1):128–142, 2005.

[77] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[78] Olivier Coudray, Christine Keribin, Pascal Massart, and Patrick Pamphile. Risk bounds for PU learning under Selected At Random assumption. *arXiv preprint arXiv:2201.06277*, 2022.

[79] Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. Dist-PU: Positive-unlabeled learning from a label distribution perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14461–14470, 2022.

[80] Ingo Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18(3):768–791, 2002.

[81] Issam Dagher. Quadratic kernel-free non-linear support vector machine. *Journal of Global Optimization*, 41(1):15–30, 2008.