New Mathematical and Computational Methods for Machine Learning and Multi-Objective Reinforcement Learning

Francois Buet-Golfouse

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Mathematics

University College London

February 8, 2024

I, Francois Buet-Golfouse, 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

This thesis concerns various aspects of robustness in machine learning, which refers broadly to the impact of certain modelling assumptions on a model's quality. The topic is examined from various perspectives, ranging from theoretical statistics to applied modelling. The main message of the six chapters is that problem framing and the probabilistic properties of algorithms used in data science are crucial for inferring robust insights from data and models.

The first part discusses two machine learning applications: classification and recommender systems. It answers an open question about the type of aleatoric uncertainty supporting the principle of margin maximisation for classification, establishing that heavy-tailed distributions do not fit this framework under certain conditions. For recommender systems, the focus is on designing a flexible method that can be applied to a range of use cases while being economical in terms of parameters, particularly for small and large datasets.

The second part focuses on fairness in machine learning, exploring situations where there are trade-offs between objectives such as accuracy and equal opportunities for different groups based on protected characteristics. A framework based on probably approximately correct learning is proposed to address the challenge of generalising to new data, and group functionals are suggested as a simple approach to fairness in unsupervised learning algorithms.

The third and final part considers situations where agents must optimise multiple conflicting objectives. New reinforcement learning and multi-armed bandit algorithms are proposed, allowing agents to learn policies over the space of trade-offs. The thesis also explores the idea that an agent's preferences may change over time, which is particularly relevant to economic and financial problems such as optimal trade execution.

Impact Statement

The research presented in this thesis addresses critical issues in machine learning from the perspectives of robustness, fairness, and multi-objective optimisation. The work offers novel insights into the underlying principles of machine learning algorithms and provides practical solutions to mitigate the negative impact of biased or suboptimal models on individuals and communities.

The first part of the thesis focuses on robustness in machine learning, specifically in the context of margin maximisation in classification algorithms. The work highlights the importance of tail distributions and loss function properties in this optimisation process and shows that margin maximisation applied to heavy-tailed distributions can lead to suboptimal performance. The thesis also introduces a new class of models, kernel factorisation machines, which can effectively fit flexible recommender systems with fewer parameters, addressing both small- and big-data contexts.

The second part of the thesis deals with fairness in machine learning, which is essential for building ethical and trustworthy systems that can benefit everyone equally. The research proposes a disciplined methodology for tackling bias in unsupervised learning algorithms and investigates the challenges of checking machine learning algorithms for fairness and correcting possible biases. The work also provides theoretical guarantees on out-of-sample generalisation, thus ensuring that the proposed algorithms can be applied to real-world problems with confidence.

The final part of the thesis explores multi-objective optimisation in machine learning, which is crucial for real-life situations where multiple (often conflicting) objectives must be considered. The research proposes new reinforcement learning and multi-armed bandit algorithms, which enable agents to learn policies over the space of trade-offs. The work also presents a new perspective whereby an agent's preferences (e.g., utility function with multiple inputs) can change through time.

Overall, the research presented in this thesis has several important academic and practical implications. From an academic perspective, the work contributes to the fields of machine learning, statistics, and data science by providing novel insights into the underlying principles of algorithms and proposing new models and methods for solving complex problems. The research also provides theoretical guarantees and practical algorithms that can be used to build more robust, fair, and effective machine learning systems.

From a practical perspective, the research has important implications for a wide range of industries, including finance, healthcare, and e-commerce. For example, the proposed recommender systems can improve user experience, increase engagement, and drive revenue for businesses, while the multi-objective optimisation algorithms can help investors optimise their portfolios based on multiple objectives, such as minimising risk, maximising returns, and meeting certain ethical or environmental criteria. The fairness considerations can also help mitigate the negative impact of biased algorithms on individuals and communities and ensure that machine learning systems are built ethically and responsibly.

In summary, the research presented in this thesis has the potential to significantly advance the field of machine learning and contribute to building more ethical, fair, and effective machine learning systems that can benefit individuals and communities alike.

Acknowledgements

No work in this thesis would have been possible without the support and benevolence of my supervisor, Professor Andrea Macrina, to whom I am incredibly indebted for accepting to take me on as a part-time PhD student. His guidance, encouragement and candour have helped turn ideas and intuitions into publications. I have learnt from him to start small but think big. I would also like to thank Professor Steven Bishop for accepting to be my subsidiary supervisor and for nudging me over the course of this journey. Furthermore, I am very honoured that Professor Jinghao Xue took the time to read a preliminary draft of this thesis and made many insightful comments. Last, I am very grateful to Professors Hao Ni and Harald Oberhauser for accepting to examine this thesis.

I am also very thankful to my managers, first and foremost Dr Anthony Owen and Dr Alistair McLeod, for their support over the years and for giving me some flexibility to attend classes and conferences. They are also the ones who first showed me that applied mathematics could solve all sorts of vital problems. My interest in research was encouraged by Professors Jean-Michel Lasry, Pierre-Louis Lions and Sylvie Méléard. With much delay, this thesis is also the result of our discussions and your kind words. Dr Jiahao Chen was instrumental in my much-improved understanding of fairness and ethics in artificial intelligence while my work on sustainable artificial intelligence and its practical implementation benefited greatly from conversations with Sinead Connolly and Dr Amit Mehta.

Last, but not least, I am indebted to my co-authors, friends, work family and loved ones for their continuous encouragement, interest in my work and patience. They will recognise themselves (and spare me the embarrassment of forgetting one of them in a long list...). I would also like to extend my recognition to all the baristas who have let me work for hours in their cafés...

This thesis is dedicated to my mother's unwavering support ("R.A.A.") and my father's memory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

To achieve robust and high-quality machine learning algorithms, it is essential to consider a range of factors that impact the models' reliability, accuracy, and effectiveness. These factors are complementary and interconnected. They include the mathematical analysis of algorithmic assumptions, the ability to develop models that can handle diverse situations and scenarios, and the reflection of the multi-objective nature of many real-world problems. Together, these components enable the development of more resilient, accurate, and effective algorithms that can address the challenges posed by real-world applications.

One such factor is the mathematical analysis of limitations linked to the algorithms' underlying assumptions (Lattimore & Szepesvári, 2020; Vapnik, 1998). By identifying and characterising these assumptions, a better understanding of the circumstances under which the algorithms are likely to perform well, and those where they may be prone to errors or limitations can be gained. Such analysis can lead to the development of more resilient and top-performing algorithms that are less sensitive to specific assumptions.

Another essential aspect of achieving quality in machine learning is finding models that can handle multiple situations and scenarios (Bousquet & Elisseeff, 2002). Machine learning algorithms are typically developed and tested on specific datasets, and their performance can be significantly affected by changes in the input data. Therefore, developing models capable of generalising well to new and unseen data is essential. This requires a multi-faceted approach that includes selecting appropriate features, using regularisation techniques, and applying methods that account for real-world data's inherent variability and complexity.

A third aspect contributing to resilience in machine learning is reflecting more closely the multi-objective nature of many problems (Glukhov, 2022; Silver et al., 2021; Vamplew et al., 2022). In many real-world applications, the objective is to maximise accuracy and consider factors such as interpretability, fairness, and privacy (Pfisterer, 2022). Similarly, in finance, many use cases include weighing the risk and reward of a specific action (Dixon et al., 2020). Therefore, it is necessary to develop models that can balance multiple objectives simultaneously. This requires a deeper understanding of the trade-offs between different objectives and developing models that can account for these trade-offs principally and effectively.

This thesis discusses robustness in machine learning and examines it from various angles, concluding that the problem framing and probabilistic properties of algorithms are crucial for obtaining robust insights. It is divided into three parts, covering classification and recommender systems, fairness in machine learning, and optimising multiple conflicting objectives. The thesis proposes new algorithms and frameworks to address these challenges. Next, the main challenges are highlighted.

1.2 Challenges

1.2.1 Robustness in classification

Robustness in machine learning refers to the ability of a model to maintain its performance even when the data it is trained on or tested with is corrupted, noisy, or perturbed. Robustness is an essential property of machine learning models, as real-world data is often subject to various types of perturbations, such as missing or incorrect values, adversarial attacks, or changes in the data distribution (Xu & Mannor, 2012).

There are several approaches to improving the robustness of machine learning models, including data augmentation, regularisation, adversarial training, and model distillation. Data augmentation involves generating new training data by applying random transformations or perturbations to the existing data, which can help the model learn to handle different types of variations in the data (Bhagoji et al., 2018). Regularisation methods, such as L1 or L2 regularisation, penalise complex models to prevent overfitting the training data and improving generalisation to new data (Hastie et al., 2015a). Adversarial training involves training the model with adversarial examples and carefully crafted inputs to fool the model into making incorrect predictions (Ganin et al., 2016; Goodfellow et al., 2016). Model distillation involves training a smaller, simpler model to mimic the predictions of a larger, more complex model, which can improve the robustness of the smaller model to perturbations in the data (Polino et al., 2018).

Data-driven approaches have become increasingly popular in many fields, as modellers are tasked with developing solutions to complex questions by selecting appropriate datasets and models to link inputs to desired outputs. In this process, modellers must select an appropriate loss function that measures the discrepancy between predicted and actual outcomes and is to be minimised (Goodfellow et al., 2016; Hastie et al., 2009). While this choice may initially seem secondary, recent research has highlighted the importance of carefully selecting a loss function appropriate for the underlying distribution of the data.

This thesis focuses on binary classification tasks that aim to predict whether an observation belongs to one of two classes. Specifically, the principle of margin maximisation is examined as a core principle that underpins many classification algorithms in data science, such as logistic regression or support vector machines. Margin maximisation involves producing positive margins as often as possible and can greatly impact the accuracy of a model's predictions (Mangasarian, 1999; Vapnik, 1998). Therefore, a key question to address is how to assess the suitability of margin maximisation for a given dataset and a particular task (Rosset et al., 2003, 2004). The first chapter explores practical criteria for evaluating margin maximisation in binary classification tasks, ultimately highlighting the crucial role of selecting an appropriate loss function in the success of data-driven approaches.

1.2.2 Recommender systems

Recommender systems are a type of algorithmic system designed to predict user preferences or interests and provide recommendations for items or services that are likely to be of interest to the user (Schafer et al., 1999). These systems are widely used in various online platforms, such as e-commerce sites, social networks, and entertainment services, to improve the user experience and increase engagement.

There are several types of recommender systems, including collaborative filtering, content-based filtering, and hybrid systems (Aggarwal et al., 2016; Burke, 2002). Collaborative filtering algorithms analyse the user's past behaviour, such as purchase history or ratings, and find other users with similar preferences to make recommendations (Sarwar et al., 2001). Content-based filtering algorithms analyse the content of the items, such as text, images, or audio, and recommend items with similar attributes to the ones the user has previously shown interest in (Van Meteren & Van Someren, 2000). Hybrid systems combine both collaborative and content-based filtering to provide more accurate and diverse recommendations (Basilico & Hofmann, 2004). Some of the popular examples of recommender systems include Amazon, which uses collaborative filtering to recommend products, Netflix, which uses a combination of collaborative and content-based filtering to recommend movies and TV shows; and Spotify, which uses a combination of collaborative filtering, content-based filtering, and natural language processing to recommend songs and playlists to users. Other examples include YouTube, LinkedIn, TripAdvisor, and Pandora.

Recommender systems have been an active research area in machine learning and data mining for over two decades (Adomavicius & Tuzhilin, 2005). Collaborative filtering and content-based filtering are two classical approaches to recommendations. At the same time, matrix factorisation techniques, such as singular value decomposition (SVD) and its variants, have become increasingly popular in recent years (Koren, 2008). One recent development in recommender systems is the use of factorisation machines, a type of supervised learning algorithm that can be used for both classification and regression tasks. Factorisation machines are particularly well-suited to problems with large and sparse feature spaces, as they can learn low-dimensional representations of the features that capture both linear and non-linear interactions economically (Rendle, 2010). Despite the promise of factorisation machines and other advanced techniques, building effective recommender systems remains a challenging task. One major challenge is the cold-start problem, where a new user or item has no or limited historical data available for recommendations (Schein et al., 2002). Another challenge is the issue of data sparsity, where only a small fraction of the total user-item matrix is observed, making it difficult to accurately predict user preferences for unobserved items (Koren, 2008). In addition, there is growing recognition that fairness and transparency are important considerations in recommender systems, particularly in sensitive domains such as employment, housing, and finance (Buet-Golfouse & Utyagulov, 2022a). Recommender systems have been shown to reproduce and amplify biases present in the underlying data, leading to discrimination against certain groups (Ekstrand et al., 2018).

1.2.3 Fairness in machine learning

Machine learning (ML) is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. The main goal of machine learning is to find patterns in data and use them to make accurate predictions or decisions on new data (Hastie et al., 2009; Shalev-Shwartz & Ben-David, 2014). There are three main types of machine learning (Russell, 2010): supervised learning, unsupervised learning, and reinforcement learning (Sutton & Barto, 2018). In supervised learning, the algorithm is trained on labelled data, where the desired output is known for each input. The algorithm then uses this labelled data to predict new, unlabeled data. On the other hand, unsupervised learning is used when the data is unlabeled, and the algorithm is tasked with finding patterns and relationships in the data without a predefined output. Reinforcement learning is used when the algorithm interacts with an environment and learns through trial and error to maximise a reward signal.

Fairness in machine learning is a critical issue that has gained increasing attention recently (Mehrabi et al., 2021; Oneto & Chiappa, 2020). While ML can potentially revolutionise various industries, its impact on social, ethical, and legal issues cannot be ignored. Fairness in ML refers to ensuring that the outputs of ML models are not biased against individuals or groups based on sensitive attributes such as race, gender, age, or socioeconomic status (Barocas et al., 2017). Fairness is essential for several reasons (Agrawal et al., 2020). Firstly, ML models are increasingly used in decision-making processes affecting people's lives, such as lending, hiring, and criminal justice. If these models are biased, they can result in discriminatory outcomes that harm specific individuals or groups. This can lead to social injustice, increased inequality, and the perpetuation of systemic discrimination. Secondly, fairness is important for the trust and adoption of ML systems. If people perceive ML models as unfair, they may be less likely to use them, leading to decreased accuracy and efficiency in decision-making processes. This, in turn, can negatively impact the performance and effectiveness of ML models.

Despite its importance, fairness in ML is difficult to achieve for several reasons. One of the main challenges is that many ML models are complex and opaque, making it difficult to identify and correct biases. This is especially true for deep learning models, often called black boxes due to their inscrutable nature (Rudin, 2019). Additionally, the data used to train ML models can be biased, leading to biased outcomes. For example, if historical data used to train a hiring model shows bias against a certain group, the model will learn this bias and perpetuate it in its outputs (Barocas & Selbst, 2016). Fairness is also often subjective and context-dependent. What may be considered fair in one situation or culture may not be considered fair in another. This makes it difficult to define and operationalise fairness in a universally applicable way (Dwork et al., 2012). To address these challenges, several approaches have been proposed. One approach is to use fairness metrics to measure and quantify the degree of bias in ML models. This can help identify and correct biases in the model (Corbett-Davies et al., 2017). Another approach is to use diverse and representative data to train the model, which can reduce the risk of bias in the data and model (Žliobaitė, 2017). Moreover, the interpretability and explainability of an ML model can help identify the source of bias and enable human intervention. For example, counterfactual explanations can help identify the changes needed in the data or model to remove biases (Rudin, 2019).

1.2.4 Multi-objective reinforcement learning

Reinforcement learning (RL) and multi-armed bandits (MAB) are two related but distinct fields of study in machine learning. Both fields involve learning from feedback but differ in several important ways. Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment (Sutton & Barto, 2018). The agent receives feedback through rewards or punishments; its goal is to maximise the cumulative reward it receives over time. RL is commonly used in applications such as game playing, robotics, and recommendation systems (Silver et al., 2016).

On the other hand, multi-armed bandits is a simpler form of learning where an agent must choose between several actions, each of which provides a reward with some probability (Lattimore & Szepesvári, 2020). The agent aims to maximise its total reward over a finite number of trials. MAB is commonly used in online advertising, clinical trials, and portfolio management (Auer et al., 2002). One key difference between RL and MAB is the structure of the feedback. In RL, the agent receives feedback after each action, allowing it to learn from its mistakes and adjust its behaviour accordingly. In contrast, in MAB, the agent receives feedback only after each trial, which limits its ability to adapt to changing conditions. Another difference between RL and MAB is the complexity of the decision-making problem. In RL, the agent must learn a policy that maps states to actions, which can be a highly complex and challenging problem. In contrast, the decision-making problem in MAB is simpler because the agent only needs to choose between a fixed set of actions. RL and MAB share many similarities despite these differences, and the two fields often overlap. For example, some RL algorithms, such as Q-learning and SARSA[1], can be applied to MAB problems (Lattimore & Szepesvári, 2020). Similarly, some MAB algorithms, such as the upper confidence bound algorithm,

[1] State–action–reward–state–action (SARSA) is an algorithm for learning a Markov decision process policy.

can be used in RL problems (Sutton & Barto, 2018).

Multi-objective reinforcement learning (MORL) is an important field that has the potential to solve complex decision-making problems involving multiple objectives. MORL has been applied to various applications such as autonomous driving, robotics, finance, and healthcare, among others (Roijers et al., 2015). However, achieving optimal solutions in MORL is challenging due to several factors such as exploration-exploitation trade-offs, sample inefficiency, and the curse of dimensionality. Comprehensive overviews of MORL are presented in (Liu et al., 2014; Roijers et al., 2013), which cover various aspects of MORL, including algorithms, evaluation metrics, and applications. One of the key challenges in MORL is to find a good balance between the competing objectives (Vamplew et al., 2011). This problem is commonly known as the multi-objective optimisation problem and has been extensively studied in the optimisation community. Another challenge in MORL is to handle the uncertainty and partial observability of the environment (Hayes et al., 2022).

1.3 Thesis outline

This thesis is divided into three parts: machine learning, fairness and multi-objective reinforcement learning, reflecting different approaches towards building robust, fair and practical machine learning models.

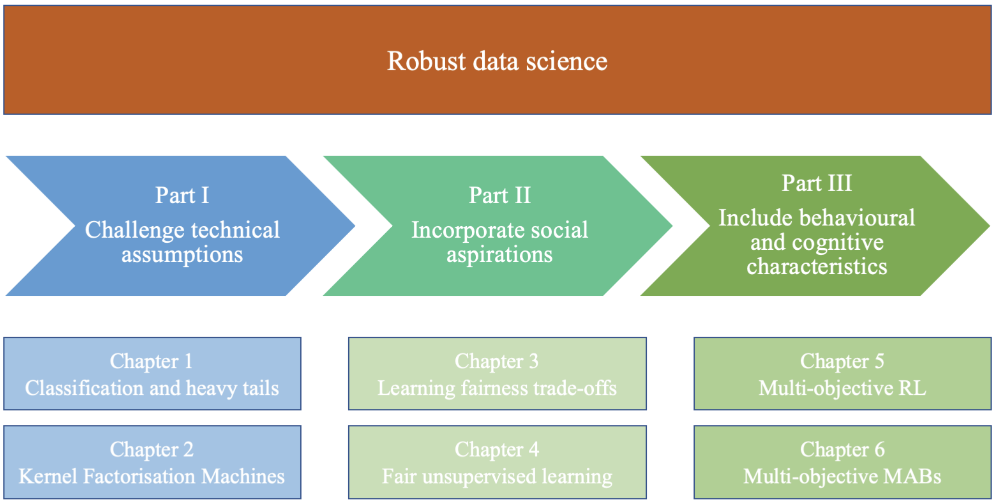
Robust data science
Part I
Challenge technical assumptions
Part II
Incorporate social aspirations
Part III
Include behavioural and cognitive characteristics
Chapter 1
Classification and heavy tails
Chapter 2
Kernel Factorisation Machines
Chapter 3
Learning fairness trade-offs
Chapter 4
Fair unsupervised learning
Chapter 5
Multi-objective RL
Chapter 6
Multi-objective MABs

Figure 1.1: Overview of the thesis research, centred around three key components.

1.3.1 Mathematical modelling in data science

In the first chapter, an important connection is established between *margin maximisation* in classification problems and the tail behaviour of loss functions. It is also shown that loss functions are closely related to the probability distribution of idiosyncratic error terms. The main contributions are twofold: first, a necessary and sufficient condition linking convergence to a margin-maximising solution and a thin-tailed requirement; second, the characterisation of loss functions that do not satisfy this requirement as regularly varying functions. The second chapter is dedicated to a new algorithm, termed “kernel factorisation machine” (“KFMs”), generalising standard factorisation machines (used, for instance, in collaborative filtering and recommender systems) via a kernel-based approach. To do so, we present a generalised representer theorem and adapt dimensionality reduction techniques to alleviate the curse of dimensionality linked to using kernels. Last, empirical results show that KFMs perform well across both small and large datasets regarding classification accuracy with a relatively small number of parameters to be fitted.

Remark 1. These results are based on the following papers:

- (Buet-Golfouse, 2021c): Buet-Golfouse, F. Narrow margins: Classification, margins and fat tails. *International conference on machine learning*. PMLR. 2021, 1127–1135;
- (Buet-Golfouse, 2021b): Buet-Golfouse, F. Asymmetry and heavy tails: Built-in robustness in classification. *International conference on learning representations. Robustml workshop*. 2021. https://sites.google.com/connect.hku.hk/robustml-2021/accepted-papers/paper-049;
- (Buet-Golfouse & Utyagulov, 2021): Buet-Golfouse, F., & Utyagulov, I. Kernel factorisation machines. *2021 20th IEEE international conference on machine learning and applications (ICMLA)*. 2021, 1748–1753;
- (Buet-Golfouse & Utyagulov, 2022a): Buet-Golfouse, F., & Utyagulov, I. Towards fair multi-stakeholder recommender systems. *Adjunct proceedings of the 30th ACM conference on user modeling, adaptation and personalization*. 2022, 255–265.

1.3.2 Fair machine learning

The third chapter is dedicated to another problem in classification, namely the *fairness* of classification algorithms. It investigates the notion of fairness trade-offs in machine learning from a *Probably approximately correct* point of view. It focuses on *partial* debiasing, thus accounting for the usual bias-accuracy trade-off. While it guarantees the learnability of these trade-offs, it also points to the role of class imbalance. These insights are key in designing policies that mitigate bias in algorithms. I then proceed to a use case building on these findings proposing a new framework to tackle bias in unsupervised learning. This is achieved by considering generalised low-rank models and modifying them by introducing a so-called “fairness functional”, which helps modellers tune the cost of unfairness across groups. In line with the previous chapter, I consider out-of-sample properties of such algorithms and show empirically that full debiasing is not always preferable due to generalisation issues.

Remark 2. These results are based on the following papers:

- (Buet-Golfouse & Utyagulov, 2023): Buet-Golfouse, F., & Utyagulov, I. Fairness trade-offs and partial debiasing. *Asian conference on machine learning*. PMLR. 2023, 112–136;

- (Buet-Golfouse & Utyagulov, 2022b): Buet-Golfouse, F., & Utyagulov, I. Towards fair unsupervised learning. *2022 ACM conference on fairness, accountability, and transparency*. 2022, 1399–1409;

- (Buet-Golfouse & Utyagulov, 2022a): Buet-Golfouse, F., & Utyagulov, I. Towards fair multi-stakeholder recommender systems. *Adjunct proceedings of the 30th ACM conference on user modeling, adaptation and personalization*. 2022, 255–265;

- (Buet-Golfouse, 2020): Buet-Golfouse, F. Partially aware: Some challenges around uncertainty and ambiguity in fairness. *Advances in neural information processing systems. Workshop on fair AI in finance*. 2020. https://sites.google.com/view/faif2020/paper-download.

1.3.3 Multi-objective reinforcement learning

Finally, the third part deals with multi-objective reinforcement learning. The fifth chapter discusses multi-objective reinforcement learning (MORL) in scenarios where an agent aims to achieve multiple tasks with competing goals but lacks complete knowledge of how to balance them. A new approach is proposed by considering the dynamics of preferences over tasks, which leads to a straightforward approach involving a surrogate state space of both states and preferences. This allows the development of effective deep Q-learning and actor-critic methods for learning multi-dimensional value functions under a preference-dependent policy. The last chapter proposes a new framework for solving multi-objective multi-armed bandit problems, where multiple rewards are considered simultaneously. The framework uses Gaussian Processes, which offer a flexible approach for modelling the overall reward function and extends prior work on the Gaussian Process Upper Confidence Bound algorithm to the multi-objective setting. One of the strengths of this approach is its ability to handle varying levels of observability.

Remark 3. These results are based on the following papers:

- (Buet-Golfouse & Pahwa, 2023): Buet-Golfouse, F., & Pahwa, P. Robust multi-objective reinforcement learning with dynamic preferences. *Asian conference on machine learning*. PMLR. 2023, 96–111;

- (Buet-Golfouse & Hill, 2023): Buet-Golfouse, F., & Hill, P. Optimal execution via multi-objective multi-armed bandits. *Proceedings of the AAAI conference on artificial intelligence. 37*. 2023.

1.4 Further contributions and publications

While not part of the present thesis, additional research undertaken and published during this PhD furthered some of the topics emphasised in this work.

1.4.1 Quantitative finance

Indeed, research in quantitative finance led to five publications. First, efforts on modelling credit portfolio risk measures with Hermite polynomials were concluded in (Buet-Golfouse & Owen, 2016) with Anthony W. Owen. Second, biconvex optimisation techniques were explored in (Buet-Golfouse et al., 2022a) to tackle specific sparse portfolio allocation problems. A new approach to pricing capital transfer transactions was presented in (Buet-Golfouse, 2017). Improving the convergence of neural partial differential equation solvers was the topic of (Buet-Golfouse et al., 2023). Last, kernel theory was applied to Volterra diffusions to "lift" them and produce an approximate Markovian representation thereof in (Buet-Golfouse & Martin, 2023).

- (Buet-Golfouse & Owen, 2016): Buet-Golfouse, F., & Owen, A. (2016). The application of Hermite polynomials to risk allocation. *Journal of Risk*, *18*(3), 77 –110;

- (Buet-Golfouse et al., 2022a): Buet-Golfouse, F., Roggeman, H., & Utyagulov, I. Rayleigh portfolios and penalised matrix decomposition. *Companion proceedings of the web conference 2022*. 2022, 579–582;

- (Buet-Golfouse, 2017): Buet-Golfouse, F., & Utyagulov, I. Kernel factorisation machines. *2021 20th IEEE international conference on machine learning and applications (ICMLA)*. 2021, 1748–1753;

- (Buet-Golfouse et al., 2023): Buet-Golfouse, F., Utyagulov, I., Pahwa, P., & Hill, P. Turbo-charging deep learning methods for partial differential equations. *Proceedings of the fourth acm international conference on AI in finance*. 2023, 150–158;

- (Buet-Golfouse & Martin, 2023): Buet-Golfouse, F., & Martin, N. W. Lifting Volterra diffusions via kernel decomposition. *Proceedings of the fourth acm international conference on AI in finance*. 2023, 481–489.

1.4.2 Robutness and simplicity

Additional research on RNA prediction via robust collaborative learning gave rise to state-of-the-art performance on RNA sequence classification tasks and resulted in (Buet-Golfouse et al., 2022b). Reflections on a novel Bayesian approximation technique dubbed the "Local Laplace" method and related to the so-called integrated nested Laplace approximation (Rue et al., 2009) led to (Buet-Golfouse & Roggeman, 2022).

- (Buet-Golfouse et al., 2022b): Buet-Golfouse, F., Roggeman, H., & Utyagulov, I. Robust collaborative learning for sequence modelling. *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2022, 1146–1150;

- (Buet-Golfouse & Roggeman, 2022): Buet-Golfouse, F., & Roggeman, H. Numerical approximations of log Gaussian cox process. *Proceedings of the AAAI conference on artificial intelligence*. *36*. (11). 2022, 12923–12924.

1.4.3 Fairness under uncertainty

Additional research addressed challenges when implementing fairness-aware models and algorithms in real-world machine learning applications, specifically in uncertainty and ambiguity. Three problems were considered: 1) models are situated in a broader decision-making context, which requires careful bias attribution; 2) partial observation of protected attributes; and 3) testing for bias with limited access to statistics. A first paper used convex optimisation to examine lower- and upper-bounds on fairness metrics (Utyagulov et al., 2023). In contrast, a second paper introduced fairness and causality decomposition formulas (Hill & Buet-Golfouse, 2023) to help practitioners diagnose possible sources of bias or confounding effects in their models.

- (Utyagulov et al., 2023): Utyagulov, I., Buet-Golfouse, F., & Hill, P. Fairness under partial observability (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K. OpenReview.net, 2023. https://openreview.net/pdf?id=if1Mmrxf-pq;

- (Hill & Buet-Golfouse, 2023): Hill, P., & Buet-Golfouse, F. Decomposing causality and fairness (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K. OpenReview.net, 2023. https://openreview.net/pdf?id=Lm7z2vYergk.

1.4.4 Human-AI collaboration

Finally, a position paper dealing with Human-Centred Artificial Intelligence (Buet-Golfouse, 2021a) discussed the limitations of purely data-driven approaches in healthcare and proposed a hybrid expert system. Domain experts possess valuable knowledge and insights that can aid in improving the accuracy and relevance of machine learning (ML) models. Incorporating expert opinions can enable the models to capture important nuances and factors that may not be evident in data-driven methods alone. Integrating ML models with human experts has become increasingly common in real-world applications. A Bayesian framework for human-in-the-loop pipelines in which the final decision combines algorithm and expert opinions, with deferral systems being a particular case. The proposed framework includes a method for updating expert opinion priors with information sharing between experts, which is a key factor in achieving superior performance (Pahwa et al., 2023). Last, the role of humans in ecosystems and its impact on renewable resources was explored in (Martin et al., 2023).

- (Buet-Golfouse, 2021a): Buet-Golfouse, F. ‘‘Art meets science”: Tackling data and perceptions. *KDD ’21: The 27th ACM SIGKDD conference on knowledge discovery and data mining*.

Workshop on understanding public perceptions for applied data science. 2021. https://dl.acm.org/doi/10.1145/3447548.3469459;

- (Pahwa et al., 2023): Pahwa, P., Thakur, K., & Buet-Golfouse, F. Dynamic human AI collaboration (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K. OpenReview.net, 2023. https://openreview.net/pdf?id=Muwb2KohnX;
- (Martin et al., 2023): Martin, N. W. D., Hill, P., Tan, T. S., & Buet-Golfouse, F. Sustainable resource management (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K. OpenReview.net, 2023. https://openreview.net/pdf?id=DLwlmWwmJBi.

Part I

Robustness in Machine Learning

Chapter 2

Narrow Margins and Heavy Tails

Research Objectives

When modellers are faced with a question and tasked with developing a data-driven approach to answer it, they typically have to select a dataset and choose a model to link inputs to the desired output. Another choice they have to make, however, is the type of loss function, which is to be minimised, measuring the discrepancy between the predicted and actual outcomes. While this may seem like a secondary choice, this chapter shows that it is actually crucial and, significantly, depends on the data's underlying distribution. In this thesis, the focus lies on binary classification tasks (i.e., when trying to predict whether an observation belongs to one class or another) and, more precisely, on the principle of margin-maximisation. In a classification task with a response y in $\{-1,+1\}$ and model $f(x)$, the margin, defined as $m := yf(x)$, is analogous to the residuals $y - f(x)$ in a regression setting. Furthermore, the classification rule $h(x) = \text{sign}(f(x))$ implies that observations with a positive margin are classified correctly and those with a negative margin incorrectly. Thus, the overarching objective of a classification problem is to produce positive margins as often as possible. Margin-maximisation can be understood as a core principle that underpins many classification algorithms in data science, such as logistic regressions or support vector machines. A key question to address is thus that of practical criteria to assess margin-maximisation's suitability on a given dataset and for a particular task, which is investigated here.

2.1 Introduction

Margin-maximisation in machine learning. Margin-maximisation (Hastie et al., 2009; Vapnik, 1998) is a fundamental concept in machine learning that aims to identify the optimal hyperplane to separate the data into distinct classes with the maximum possible margin. The margin is defined as the distance between the hyperplane and the closest data points, and maximising it can lead to better generalisation performance of the resulting classifier. Margin-maximisation is an important technique in many popular machine learning algorithms, such as support vector machines ("SVM), boosting, and perceptron algorithms. SVMs, in particular, are well known for their ability to find the optimal hyperplane that maximizes the margin, which can result in excellent performance in classification tasks.

The margin can be seen as a measure of confidence in the classifier's predictions. A larger margin indicates that the classifier is more confident in its predictions, as there is a larger distance between the decision boundary and the closest data points. This can help to reduce the risk of overfitting and improve the generalisation performance of the classifier. The concept of margin-maximisation has also been extended to other machine learning tasks, such as regression and clustering. In regression tasks, margin maximisation aims to find a hyperplane that minimises the distance between the predicted and actual values, while in clustering tasks, margin maximisation aims to find a hyperplane that separates the clusters with the maximum possible margin.

However, as the sample size grows, this property is less relevant because the dataset is unlikely to be separable. Margin-maximisation and separating hyperplanes are still appealing for (at least) two reasons: first, they are an intuitive concept and are a benchmark in any classification task, and second, using boosting or kernel support vector machines ("SVM") in a higher dimensional space can enable separability.

Convergence of regularised problems to margin-maximising solutions. (Rosset et al., 2003, 2004), building on previous work by (Bartlett et al., 2006; Freund & Schapire, 1997; Friedman et al., 2000; Schapire et al., 1997) and (Mangasarian, 1999), consider the case of a linear classifier (e.g., logistic regression or support vector machine) and investigate the convergence of a regularised estimator to a margin-maximising hyperplane when data is separable. Intriguingly, the authors established that under an apparently mild criterion on the loss function, this convergence was guaranteed. This was an important result from a couple of standpoints. First, it established a relationship between regularised classifiers and margin-maximisation, and second, it showed that usual loss functions shared that property, leading to the exact choice of a link function being of second order.

Margin-maximisation and probabilistic properties. The key results of the work in this thesis are the (partial) answer to the open question and conjecture in (Rosset et al., 2003), on the one hand, and the link between the non-convergence to a margin-maximising classifier and regular variation (Bingham et al., 1989) of the loss function, on the other hand. While margin-maximisation is quite specific to the linear setting, deriving analytical properties of loss functions used in other settings, such as deep learning, is particularly interesting to understand choices for loss functions and their implications.

A connection between this problem and heavy tails (i.e., probability distributions whose tails are not exponentially bounded) is established in this thesis; (Taleb, 2020) offers a wide-ranging review of heavy tails in multiple applications, (Ibragimov et al., 2015) consider more specifically the role of heavy tails in finance and inference, and applications in supervised learning (mainly regression) are described in (Brownlees et al., 2015; Hsu & Sabato, 2016; Lugosi & Mendelson, 2019). Earlier approaches such as (Chatterjee & Hadi, 1986; Huber & Ronchetti, 2009; Wang et al., 2007) considered heavy tails through the lens of robust estimation. Additional research in classification tasks under a heavy-tail regime is warranted to refine the current state of understanding.

Contributions. The contributions in this chapter can be articulated around three questions:

- *Is there a converse statement to (Rosset et al., 2003)'s sufficient condition?* In other words, must the loss function satisfy the same criterion if a normalised and regularised estimator converges to a margin-maximising hyperplane? It is shown that this indeed holds under some additional assumptions, namely the convexity and differentiability of the loss function ℓ.
- *If the ratio criterion is not verified, what can be said about the loss function?* Interestingly, it is established that such losses can be shown to be in the class of regularly varying functions under mild assumptions (see (Bingham et al., 1989) for an introduction to the theory of regularly varying functions).
- *Is there a probabilistic interpretation of these analytical results?* Using the latent interpretation of binary classification models, it is proved that the distribution of the latent variable must also be regularly varying, loosely characterised by heavy tails.

While the starting point of this work in (Rosset et al., 2003) has to do with linear models and margin-maximising solutions, the characterisation of loss functions (and behaviour thereof) is of broad interest.

Notations and definitions used in the chapter. To consider the case of binary classification, suppose that there are n observations of a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ and group label $y_i \in \{-1, 1\}$ (which is the object to be classified), for $i = 1, ..., n$. The loss function $\ell : \mathbb{R} \to \mathbb{R}^+$ depends only on the margin, is monotonic, non-increasing, non-negative and continuous, while the underlying model $g(\mathbf{x}) = \beta^T h(\mathbf{x})$ is taken to be linear. As usual (Shalev-Shwartz & Ben-David, 2014), the empirical risk is minimised:

$$\min_{\beta \in \mathbb{R}^{|\mathscr{H}|}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i \beta^T h(\mathbf{x}_i)), \tag{2.1}$$

where $\mathscr{H} = \{h_1(\mathbf{x}), \cdots\}$ is an H-dimensional dictionary of functions from $\mathbb{R}^d$ to $\mathbb{R}$ and $\beta \in \mathbb{R}^H$ is the vector of weights. The prediction at point $\mathbf{x}$ is simply $\operatorname{sign}\left(\beta^T h(\mathbf{x})\right)$. But, as pointed out by (Rosset et al., 2003), when $H = |\mathscr{H}|$ is large, some regularisation is required to be able to control the complexity of the classifier:

$$\min_{\beta \in \mathbb{R}^{|\mathscr{H}|}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i \beta^T h(\mathbf{x}_i)) + \lambda \|\beta\|_p^p, \tag{2.2}$$

for $p \geq 1$, where $\lambda \geq 0$ is a non-negative regularisation parameter. In the following, β_λ denotes (possibly one of) the solution(s) to the optimisation problem in Equation (2.2).

2.2 The sufficient condition

Let us start by recalling the main result from (Rosset et al., 2003), before suggesting what this theorem means in practice in a remark.

Theorem 1. *(Theorem 2.1 in (Rosset et al., 2003)) Assume that the data* $\{\mathbf{x}_i, y_i\}_{i=1}^n$ *is separable (i.e., there exists* $\beta \in \mathbb{R}^{\mathscr{H}}$ *such that* $\min_i y_i \beta^T h(\mathbf{x}_i) > 0$ *(note that* $m_i := y_i \beta^T h(\mathbf{x}_i)$ *is data point i's margin). Let* ℓ *be a monotone non-increasing, non-negative loss function depending on the margin only. If there exists* $T > 0$ *(possibly* $T = +\infty$*) such that*

$$\lim_{t \to T} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = +\infty, \tag{2.3}$$

for all $\varepsilon \in (0,1)$*, then* ℓ *is a margin-maximising loss function in the sense that any convergence point of the normalised solutions* $\frac{\beta_\lambda}{\|\beta_\lambda\|_p}$ *to the regularised problem (Eq. (2.2)) as* $\lambda \to 0$ *is an* L^p *margin-maximising separating hyperplane. Consequently, if the margin-maximising hyperplane is unique, then the solutions converge to it*

$$\lim_{\lambda \to 0} \frac{\beta_\lambda}{\|\beta_\lambda\|_p} = \operatorname*{arg\,max}_{\beta, \|\beta_\lambda\|_p = 1} \min_i y_i \beta^T h(\mathbf{x}_i). \tag{2.4}$$

Remark 4. The condition $\lim_{t \to T} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = +\infty$ has a natural interpretation. For the limit ratio condition in Eq. (2.3) to hold, it must be that $\lim_{t \to T} \ell(t) = 0$ (otherwise the ratio would be finite; the case $\lim_{t \to +T} \ell(t) = +\infty$ implies that $\ell(t) = +\infty$ for all t given the non-increasingness of ℓ). Now, if ℓ is differentiable and ℓ' is non-zero in a neighbourhood of T, then, by L'Hôpital's rule, it follows that

$$\lim_{t \to T} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = (1-\varepsilon) \lim_{t \to T} \frac{\ell'(t(1-\varepsilon))}{\ell'(t)}, \tag{2.5}$$

so that $\lim_{t \to T} \frac{-\ell'(t)}{-\ell'(t(1-\varepsilon))} = 0$. In other words, the *marginal utility* of having a margin of size t versus a margin of size $t(1-\varepsilon)$ goes to 0. Roughly speaking, this means that data points with a smaller margin contribute more to the average empirical loss (Eq. 2.1).

2.2.1 Common loss functions

It is straightforward to verify that the usual loss functions verify the criterion Eq. (2.3), such as the exponential loss function $\ell_{\text{Exponential}} : t \mapsto e^{-t}$ (used implicitly in the AdaBoost algorithm proposed (Freund & Schapire, 1997; Friedman et al., 2000)), the log-likelihood $\ell_{\text{Logistic}} : t \mapsto \log(1+e^{-t})$ used in logistic regression, or the hinge loss $\ell_{\text{SVM}} : t \mapsto \max(0, 1-t)$, which is central to support vector machines (see (Hastie et al., 2009; Vapnik, 1998)). The case $T < +\infty$ is only of interest for hinge-type losses where a cut-off is applied.

2.2.2 The case of Probit regression

It can also be shown that another well-known loss function, namely the one used in Probit regression, is not considered in (Rosset et al., 2003), which satisfies the criterion Eq. (2.3). In the case of Probit regression, the associated margin loss function is defined as $\ell_{\text{Probit}} : t \mapsto -\log(\Phi(t))$, where Φ is the standard Gaussian cumulative distribution function. Since $\lim_{t\to+\infty} \Phi(t) = 1$ and $\Phi'(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$, it comes that $\lim_{t\to+\infty} \frac{\ell_{\text{Probit}}(t(1-\varepsilon))}{\ell_{\text{Probit}}t)} = (1-\varepsilon)\lim_{t\to+\infty} e^{\frac{t^2}{2}(1-(1-\varepsilon)^2)} = +\infty$, again by L'Hôpital's rule.

Remark 5. Theorem 1, combined with the fact that the most frequently used loss functions verify the criterion in Eq. (2.3), means that if the data is separable and the margin-maximising hyperplane is unique, then the exact choice of loss function does not matter as all usual loss functions lead to the same end result. The overall results and considerations in Section 2.6.3 somewhat qualify that statement.

2.3 The necessary condition

In this Section, the goal is to answer an open question in (Rosset et al., 2003) around the existence of a converse to Theorem 1. In other words, if $\lim_{\lambda\to 0} \frac{\beta_\lambda}{\|\beta_\lambda\|_p} \to \beta_*$, where β_* is a margin-maximising hyperplane with unit norm, is it true that $\lim_{t\to T} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = +\infty$ for all $\varepsilon > 0$?

A partial positive answer to the question is proposed by only focusing here on the case where $T = +\infty$ and $p = 2$, and making the additional assumptions that ℓ is decreasing with $\lim_{t\to+\infty} \ell(t) = 0$, convex and differentiable with continuous derivative ℓ'. The loss function to be minimised is thus

$$L(\beta;\lambda) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i \beta^T h(\mathbf{x}_i)\right) + \lambda \beta^T \beta. \tag{2.6}$$

This Section goes through several steps that were taken to reach the result and starts from the assumption that the normalised "ridged" solution $\beta_\lambda/\|\beta_\lambda\|_2$ converges to a margin-maximising hyperplane β_* with unit norm.

First, notice that an expression for the normalised regularised solution vector $\beta_\lambda/\|\beta_\lambda\|_2$ can be given as a linear combination of the feature vectors $h(\mathbf{x}_i)$, for $i = 1, \cdots, n$.

Proposition 1. *For a given $\lambda > 0$, the normalised solution vector $\beta_{\lambda,1} = \frac{\beta_\lambda}{\|\beta_\lambda\|_2}$ can be expressed as*

$$\beta_{\lambda,1} = K_\lambda \sum_{i=1}^{n} \alpha_{i,\lambda} y_i h(\mathbf{x}_i),$$

where $\alpha_{i,\lambda} = \frac{\ell'(m_{i,\lambda})}{\sum_{j=1}^{n} \ell'(m_{j,\lambda})} \geq 0$ for all i, and $K_\lambda > 0$ is a normalising constant such that $\|\beta_{\lambda,1}\|_2 = 1$. In addition, the following inequality holds $0 < \frac{1}{\sqrt{n}} \min_{i=1,\cdots,n} \|h(\mathbf{x}_i)\|_2 \leq K_\lambda^{-1} \leq \max_{i=1,\cdots,n} \|h(\mathbf{x}_i)\|_2$, i.e., K_λ is always bounded by upper- and lower-bounds that are independent of λ.

Proof. The first-order condition of the problem reads

$$\frac{\partial L}{\partial \beta} = \frac{1}{n}\sum_{i=1}^{n} \ell'(m_{i,\lambda})y_i h(\mathbf{x}_i) + 2\lambda\beta,$$

leading to

$$\beta_\lambda = -\frac{1}{2\lambda n}\sum_{i=1}^{n} \ell'(m_{i,\lambda})y_i h(\mathbf{x}_i).$$

Since the loss function ℓ is decreasing, $\ell'(t) < 0$ for all $t \in \mathbb{R}$, so that $\alpha_{i,\lambda} = \frac{\ell'(m_{i,\lambda})}{\sum_{j=1}^{n}\ell'(m_{j,\lambda})}$ is positive for all i. Now, $K_\lambda^2 = \frac{1}{\|\sum_{i=1}^{n}\alpha_{i,\lambda}h(\mathbf{x}_i)\|_2^2}$; since $\sum_{i=1}^{n}\alpha_{i,\lambda} = 1$, it is well-known that $1/n \leq \sum_{i=1}^{n}\alpha_{i,\lambda}^2 \leq 1$. □

From now on, the focus is specifically on the behaviour of the weights $\alpha_{i\lambda}$.

Proposition 2. *Suppose that $h(\mathbf{x}_j)$ is not a support vector of the limiting margin-maximising hyperplane β_*, then $\alpha_{j,\lambda} \to 0$ as $\lambda \to 0$. On the other hand, if $h(\mathbf{x}_i)$ is a support vector, then $\alpha_{i,\lambda}$ is bounded by below.*

Proof. Since $\beta_{\lambda,1}$ converges to a margin-maximising hyperplane and by continuity of the minimal margin in β, this entails that there exists $\overline{\lambda} > 0$ such that for any $\lambda < \overline{\lambda}$ and for all $i = 1, \cdots, n$, $m_{i,\lambda} \geq 0$. Similarly, given that β_* corresponds to a margin-maximising hyperplane, it holds that $\beta_* = K_* \sum_{i=1}^{n} \alpha_{i,*} y_i h(\mathbf{x}_i)$, where $\alpha_{i,*} > 0$ if $h(\mathbf{x}_i)$ is a support vector (in other words, on the boundary of the slab) and $\alpha_{i,*} = 0$ otherwise (this can be obtained via the dual approach to the margin-maximisation problem, see (Vapnik, 1998) or Section 4.5.2. in (Hastie et al., 2009)).

By assumption and given the loss minimising property, the convergence of the normalised solution vector to the margin-maximising weight vector β_* as $\lambda \to 0$ follows: $\beta_{\lambda,1} \to \beta_*$, which is equivalent to $K_\lambda \alpha_{i,\lambda} \to K_* \alpha_{i,*}$. In particular, for non-support vectors, this means $\alpha_{j,\lambda} \to 0$.

It can now be established that, for any support vector $h(\mathbf{x}_i)$, its associated coefficient $\alpha_{i,\lambda}$ is bounded for λ small enough. Indeed, since $K_\lambda \alpha_{i,\lambda} \to K_* \alpha_i > 0$, then, for any $\delta > 0$, there exists λ' such that, for any $\lambda \leq \lambda'$, $\|K_\lambda \alpha_{i,\lambda} - K_* \alpha_{i,*}\|_2^2 \leq \delta$. □

This distinction between support and non-support vectors will now help characterise the behaviour of the loss function.

Proposition 3. *For any non-support vector $h(\mathbf{x}_j)$ and any support vector $h(\mathbf{x}_i)$, it holds*

$$\frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})} \to 0, \tag{2.7}$$

as $\lambda \to 0$.

Proof. Pick i such that $h(\mathbf{x}_i)$ is a support vector and j such that $h(\mathbf{x}_j)$ is *not* a support vector. Then

$$\begin{aligned} \alpha_{j,\lambda} &= \frac{\ell'(m_{j,\lambda})}{\sum_{k=1}^{n}\ell'(m_{k,\lambda})} \\ &= \frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})}\cdot\frac{\ell'(m_{i,\lambda})}{\sum_{k=1}^{n}\ell'(m_{k,\lambda})} \\ &= \frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})}\alpha_{i,\lambda}. \end{aligned}$$

Since $\alpha_{i,\lambda}$ is bounded by below, $\alpha_{j,\lambda} \to 0$ implies that $\frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})} \to 0$. □

It is now possible to characterise the limiting property and tail behaviour of the ratio of the *derivative* of the loss function.

Proposition 4. *Consider a non-support vector $h(\mathbf{x}_j)$ and a support vector $h(\mathbf{x}_i)$, with respective margins $m_{j,*}, m_{i,*}$ and let $\varepsilon \in (0, \frac{m_{j,*}-m_{i,*}}{2})$. Then*

$$\lim_{t\to+\infty}\frac{\ell'(t(1-\mu))}{\ell'(t)} = +\infty, \tag{2.8}$$

where $\mu = \frac{m_{j,*}-m_{i,*}-2\varepsilon}{m_{j,*}-\varepsilon} \in (0,1)$.

Proof. Observe that the margin can be rewritten as $m_{k,\lambda} = \|\beta_\lambda\|_2 y_k \beta_{\lambda,1}^T h(\mathbf{x}_k) := \|\beta_\lambda\|_2 \cdot m_{k,\lambda,1}$ for $k = 1,\cdots,n$. In other words, $m_{k,\lambda,1}$ is the "normalised" margin. By convergence of the normalised weight vector and by continuity of the margin, it holds that $m_{k,\lambda,1} \to m_{k,*} := y_k\beta_*^T h(\mathbf{x}_k)$. Since i is a support vector but j isn't, it comes $m_{i,*} < m_{j,*}$. Hence, for any $0 < \varepsilon < \frac{m_{j,*}-m_{i,*}}{2}$, there exists $\lambda'' > 0$ such that for all $\lambda \le \lambda''$,

$$0 < m_{i,*} - \varepsilon \le m_{i,\lambda,1} \le m_{i,*} + \varepsilon < m_{j,*} - \varepsilon \le m_{j,\lambda,1} \le m_{j,*} + \varepsilon.$$

Since ℓ is convex, it follows that ℓ' is non-decreasing, hence the key inequality:

$$0 \le \frac{\ell'(\|\beta_\lambda\|_2\cdot(m_{j,*}-\varepsilon))}{\ell'(\|\beta_\lambda\|_2\cdot(m_{i,*}+\varepsilon))} \le \frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})}. \tag{2.9}$$

But, as $\lambda \to 0$, $\|\beta_\lambda\|_2 \to +\infty$ (since for $\lambda < \overline{\lambda}$, all margins are positive but $\ell(t) > 0$, $\|\beta_\lambda\|_2$ must diverge as $\lambda \to 0$) and $\frac{\ell'(m_{j,\lambda})}{\ell'(m_{i,\lambda})} \to 0$ thanks to Proposition 3. This now implies that

$$\lim_{t\to+\infty}\frac{\ell'(t(m_{j,*}-\varepsilon))}{\ell'(t(m_{i,*}+\varepsilon))} = 0.$$

By continuity of ℓ', this is equivalent to

$$\lim_{t\to+\infty}\frac{\ell'(t(1-\mu))}{\ell'(t)} = +\infty, \tag{2.10}$$

where $\mu = \frac{m_{j,*}-m_{i,*}-2\varepsilon}{m_{j,*}-\varepsilon} \in (0,1)$. □

It now remains to derive a similar result for all $\mu \in (0,1)$ and for ℓ rather than ℓ'.

Proposition 5. *Under the assumptions of this Section, it holds*

$$\lim_{t\to+\infty} \frac{\ell(t(1-\mu))}{\ell(t)} = +\infty, \tag{2.11}$$

for any $\mu \in (0,1)$.

Proof. The result of Proposition 4 holds for any positive margins $m_{i,*}, m_{j,*}$ such that $0 < m_{i,*} < m_{j,*}$ and any $0 < \varepsilon < \frac{m_{j,*}-m_{i,*}}{2}$, hence, for any $\mu \in (0,1)$, it must hold that $\lim_{t\to+\infty} \frac{\ell'(t(1-\mu))}{\ell'(t)} = +\infty$. This is not quite the desired result, but, since $\lim_{t\to+\infty} \ell(t) = 0$ and ℓ is differentiable (and such that $\ell'(t) \neq 0$ for $t > 0$ given that ℓ is decreasing), the conditions of L'Hôpital's rule hold and it follows $\lim_{t\to+\infty} \frac{\ell(t(1-\mu))}{\ell(t)} = (1-\mu)\lim_{t\to+\infty} \frac{\ell'(t(1-\mu))}{\ell'(t)} = +\infty$, for any $\mu \in (0,1)$. □

Remark 6. Here is a simple interpretation of this Section's results. Under the assumptions made in this Section (namely $T = +\infty$, $p = 2$, ℓ is decreasing with $\lim_{t\to+\infty} \ell(t) = 0$, convex and differentiable with continuous derivative ℓ'), convergence of the normalised solution vector to a margin-maximising hyperplane implies that the ratio $\lim_{t\to+\infty} \frac{\ell(t(1-\mu))}{\ell(t)}$ must go to infinity for any $\mu \in (0,1)$. If the ratio does not go to infinity for a given $\mu \in (0,1)$, then the normalised solution vector *does not* converge to a margin-maximising hyperplane.

2.4 Functional characterisation of the loss ℓ

It has been shown that the condition on the convergence to infinity of the ratio $\ell((1-\varepsilon)t)/\ell(t)$ was a necessary (under strict assumptions) and sufficient (under mild assumptions) condition for the convergence of the normalised regularised estimator $\beta_{\lambda,1}$ to a margin-maximising solution.

In this Section, however, the interest lies in understanding the case where this ratio criterion in Eq. (2.3) does not hold and the consequences in terms of the loss function. From now on, suppose that there exists $\varepsilon \in (0,1)$, such that

$$\lim_{t\to+\infty} \frac{\ell((1-\varepsilon)t)}{\ell(t)} = \gamma \neq +\infty \tag{2.12}$$

This assumption is made, as in the previous section, that for any $a > 0$, $\lim_{t\to+\infty} \frac{\ell(at)}{\ell(t)} \in \overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{+\infty\}$, i.e., the limit exist but can be $+\infty$ or a non-negative number. This assumption is made for the sake of simplicity but can be very modified, see Section 2.4.3.

2.4.1 The ratio $\ell(at)/\ell(t)$ has a limit for all $a > 0$

The new notation $\eta := 1-\varepsilon$ is introduced.

Proposition 6. *For any* $n \in \mathbb{Z}$, $\lim_{t\to+\infty} \frac{\ell(\eta^n t)}{\ell(t)} = \gamma^n$, *where* $\eta = 1-\varepsilon$.

Proof. Given that $\lim_{t\to+\infty}\frac{\ell(\eta t)}{\ell(t)} = \gamma \geq 1$, it also holds $\lim_{t\to+\infty}\frac{\ell(t)}{\ell(\eta t)} = \frac{1}{\gamma}$, and by continuity, $\lim_{t\to+\infty}\frac{\ell(t/\eta)}{\ell(t)} = \frac{1}{\gamma}$. If one considers the case $a = \eta^n$, where $n \in \mathbb{N}^* = \mathbb{N} - \{0\}$, it is possible to write

$$\frac{\ell(at)}{\ell(t)} = \frac{\ell(\eta^n t)}{\ell(t)} = \prod_{i=1}^{n-1}\frac{\ell(\eta^{i+1}t)}{\ell(\eta^i t)}.$$

But it is observed that, by continuity, $\lim_{t\to+\infty}\frac{\ell(\eta^{i+1}t)}{\ell(\eta^i t)} = \lim_{t\to+\infty}\frac{\ell(\eta\cdot t)}{\ell(t)} = \gamma$, thus leading to $\lim_{t\to+\infty}\frac{\ell(\eta^n t)}{\ell(t)} = \gamma^n$. Bringing those two facts together, the result holds. □

Proposition 7. *There exists a function $\rho : \mathbb{R}_+^* \to \mathbb{R}_+^*$ such that*

$$\lim_{t\to+\infty}\frac{\ell(at)}{\ell(t)} = \rho(a). \tag{2.13}$$

In particular, $\rho(a) > 0$ for any positive a.

Proof. For any $0 < a < \eta$, there exists $n_a \in \mathbb{N}^*$ such that $\eta^{n_a+1} \leq a < \eta^{n_a}$, so that

$$\frac{\ell(\eta^{n_a}t)}{\ell(t)} \leq \frac{\ell(at)}{\ell(t)} \leq \frac{\ell(\eta^{n_a+1}t)}{\ell(t)}.$$

Based on our previous results (and the earlier assumption of the limit's existence in $\overline{\mathbb{R}}_+$), this implies that $\gamma^{n_a} \leq \lim_{t\to+\infty}\frac{\ell(at)}{\ell(t)} \leq \gamma^{n_a+1}$. The case $a \geq \eta$ is handled similarly, since there exists $n_a \in \mathbb{N}$ such that $\eta^{-n_a+1} \leq a < \eta^{-n_a}$. □

2.4.2 ℓ as regularly-varying function

To make use of this result, start by briefly recalling some fundamentals of regularly varying function theory (see (Bingham et al., 1989) for all results mentioned here related to regularly varying functions).

Definition 1. A (measurable) function $L : \mathbb{R}_+^* \to \mathbb{R}_+^* := \mathbb{R}_+ - \{0\}$ is said to be *slowly varying* (at infinity) if, for all $a > 0$,

$$\lim_{t\to+\infty}\frac{L(at)}{L(t)} = 1.$$

Similarly, *regularly varying* functions are introduced:

Definition 2. A (measurable) function $h : \mathbb{R}_+^* \to \mathbb{R}_+^*$ is said to be *regularly varying* (at infinity) if, for all $a > 0$,

$$\lim_{t\to+\infty}\frac{h(at)}{h(t)} = \rho(a),$$

where $\rho(a)$ is finite but non-zero for every $a > 0$.

This is precisely the setup that was established in the previous subsection, in particular in Proposition 7. One of the cornerstones of the theory of regularly varying functions is Karamata's *characterisation* theorem.

Theorem 2. *Every regularly varying function $h : \mathbb{R}_+^* \to \mathbb{R}_+^*$ is of the form*

$$h(t) = t^{\zeta} L(t),$$

where $\zeta \in \mathbb{R}$ and L is a slowly varying function.

In particular, it comes directly that $\lim_{t\to+\infty} \frac{h(at)}{h(t)} = a^{\zeta}$. This implies that the limit function ρ is uniquely defined as $\rho(a) = a^{\zeta}$ and can only be a power function. Note that a closely related result is Karamata's *representation* theorem, giving a precise representation of slowly varying functions.

In the case at hand, it can thus be concluded that if ℓ does not verify the ratio criterion, then ℓ is a regularly varying function, and it is straightforward to infer that

$$\zeta = \frac{\log(\gamma)}{\log(\eta)}. \tag{2.14}$$

Since $\eta \in (0,1)$ and $\gamma \geq 1$, then it is true that $\zeta \leq 0$. Note that $\zeta = 0$ if and only if $\gamma = 1$, in which case the loss function ℓ is slowly varying. Since ζ is non-positive, $\xi := -\zeta$ is usually considered instead of ζ directly. While the characterisation and representation of the loss function are interesting results in their own right, it is possible to make them more intuitive by adopting a probabilistic viewpoint.

2.4.3 Discussion

Before moving forward, the assumption made to obtain in Section 2.4's results can be discussed. The assumption made here is that, for any $a > 0$, $\lim_{t\to+\infty} \frac{\ell(at)}{\ell(t)} \in \overline{\mathbb{R}}_+$. This is a *global* assumption which, coupled with Eq. 2.12, implies that the limit must then be finite everywhere, i.e., $\lim_{t\to+\infty} \frac{\ell(at)}{\ell(t)} \in \mathbb{R}_+$. However, this assumption does not require any additional finiteness condition. The *global* aspect of the assumption can be significantly weakened if one posits that there exists at least another point such that the limit exists and is finite. The result, in this case, is as follows:

Theorem 3. *Let $a_1, a_2 \in \mathbb{R}_+^* - \{1\}$ such that $\frac{\log a_1}{\log a_2} \notin \mathbb{Q}$ and $\lim_{t\to+\infty} \frac{\ell(a_i t)}{\ell(t)} = \rho(a_i) < +\infty$, for $i = 1, 2$, then for any $a > 0$, $\lim_{t\to+\infty} \frac{\ell(at)}{\ell(t)} = \rho(a)$, where, for any $a > 0$, $\rho(a) = a^{\zeta}$ for some $\zeta \in \mathbb{R}$.*

Proof. Given that ℓ is non-negative and non-increasing, "Theorem K" in (Seneta, 2002) can be applied to the function g defined as $g(u) := \log \ell(e^u)$ for $u \in \mathbb{R}$. □

The condition $\frac{\log a_1}{\log a_2} \notin \mathbb{Q}$ may, however, not be obvious to check. To summarise, the main takeaway is that Eq. 2.12, on its own, is –in general– not enough to guarantee that ℓ is regularly varying and an additional assumption is required.

2.5 Probabilistic interpretation of the characterisation result

In this section, the latent interpretation of binary classification is recalled and in particular, the assumption of symmetry of the latent variable and its inherent limitation is discussed. This approach will then be applied to regularly varying losses and distributions in the next section.

2.5.1 Latent interpretation of classification

It is sometimes useful to posit a threshold model whereby a variable ε_i is unobservable but such that the observed class label $y_i \in \{-1,+1\}$ is given by

$$\mathbf{1}_{\{y_i=-1\}} = \mathbf{1}_{\{\beta^T h(\mathbf{x}_i)+\varepsilon_i<0\}}. \tag{2.15}$$

The component $\beta^T h(\mathbf{x}_i)$ is observed but the ε_i's are random perturbations (usually considered to be independent and identically distributed). This leads directly to

$$\begin{aligned}\mathbb{P}(y_i=-1|x_i,\beta) &= F(-\beta^T h(\mathbf{x}_i))\\ \mathbb{P}(y_i=+1|x_i,\beta) &= 1-F(-\beta^T h(\mathbf{x}_i)),\end{aligned}$$

with F the cumulative distribution function ("c.d.f.") of ε. In particular, under the assumption that F is symmetric around 0 (i.e., $1-F(t)=F(-t)$ for all $t\in\mathbb{R}$), then one can succinctly rewrite the probability of observing class y as

$$\mathbb{P}(y|x_i,\beta) = F(y\beta^T h(\mathbf{x}_i)), \tag{2.16}$$

for $y\in\{-1,+1\}$ and the likelihood of the sample is then

$$\mathscr{L}(\{\mathbf{x}_i,y_i\}_{i=1}^n;\beta) = \prod_{i=1}^n F(y_i\beta^T h(\mathbf{x}_i)).$$

Hence, maximising the likelihood is equivalent to minimising

$$L(\{\mathbf{x}_i,y_i\}_{i=1}^n;\beta) = \frac{1}{n}\sum_{i=1}^n -\log\left(F(y_i\beta^T h(\mathbf{x}_i))\right).$$

One can thus define in a straightforward way $\ell(t) = -\log(F(t))$. Now, given a loss function ℓ, can one find a corresponding c.d.f. F?

2.5.2 Characterisation of losses with latent interpretation

For F to be a valid c.d.f., F must be non-negative, right continuous with left limits, non-decreasing and verify $\lim_{t\to-\infty}F(t)=0$, $\lim_{t\to+\infty}F(t)=1$. These conditions are guaranteed if ℓ is continuous, non-increasing and has limit $+\infty$ in $-\infty$ and 0 in $+\infty$. However, the key assumption is that of symmetry, which is difficult to obtain.

Proposition 8. *Under the assumptions of non-negativity, non-increasingness and continuity, the loss function ℓ can be expressed as a rescaled cumulative distribution function if and only if it satisfies the* *following functional equation:*

$$2^{-\frac{\ell(t)}{\ell(0)}} + 2^{-\frac{\ell(-t)}{\ell(0)}} = 1, \tag{2.17}$$

for all $t \in \mathbb{R}$. In this case, $F(t) = e^{-\beta\ell(t)}$ with $\beta = \frac{\log 2}{\ell(0)}$.

The proof is very simple but this result is a negative one in the sense that not all loss functions can be written as $\ell(t) = -\log(F(t))$ for a symmetric F. A counterexample is the exponential loss function $\ell_{\text{Exponential}}$, leading to $F(t) = e^{-e^{-t}}$, which is a valid c.d.f. (namely, that of a Gumbel distribution) but is not symmetric.

2.6 Regularly varying latent distributions

2.6.1 Brief overview

A concept that is closely related to that of regularly varying *functions* is that of regularly varying *distributions* (cf. (Cooke et al., 2014) for an introduction to the topic), which its probabilistic equivalent insofar as it characterises the tails of distributions.

Definition 3. A cumulative distribution function F is called regularly varying at infinity with tail index $\xi \in (0, +\infty)$ if

$$\lim_{t\to+\infty} \frac{\overline{F}(at)}{\overline{F}(t)} = a^{-\xi}, \tag{2.18}$$

for any $a > 0$, where $\overline{F} = 1 - F$ is the survival function.

It is interesting to notice that for $a > 1$, $\lim_{t\to+\infty} \frac{\overline{F}(at)}{\overline{F}(t)} = \mathbb{P}(X > at | X > t)$, where $X \sim F$ (i.e., X follows the distribution given by the c.d.f F).

2.6.2 Loss function and latent distribution tail behaviours

Under the assumption that ℓ is differentiable (or equivalently that $F \simeq e^{-\ell}$ is differentiable, hence admits a probability density function f), we have

$$\begin{aligned}
\lim_{t\to+\infty} \frac{\ell(at)}{\ell(t)} &= a \cdot \lim_{t\to+\infty} \frac{\ell'(at)}{\ell'(t)} \\
&= a \cdot \lim_{t\to+\infty} \frac{F(t)}{F(at)} \cdot \frac{f(at)}{f(t)} \\
&= a \cdot \lim_{t\to+\infty} \frac{f(at)}{f(t)},
\end{aligned}$$

whence $\lim_{t\to+\infty} \frac{f(at)}{f(t)} = a^{-(\xi+1)}$. Similarly, since $\overline{F}'(t) = -f(t)$, it comes

$$\begin{aligned}
\lim_{t\to+\infty} \frac{\overline{F}(at)}{\overline{F}(t)} &= a \cdot \lim_{t\to+\infty} \frac{-f(at)}{-f(t)} \\
&= a^{-\xi},
\end{aligned}$$

In other words, it has been shown that the loss function ℓ and its associated latent distribution F have the same tail index.

Proposition 9. *If F is regularly varying with tail index ξ and is differentiable (i.e., admits a probability density function f), then f is regularly varying with tail index $\xi + 1$ and the associated loss function $\ell := -\log F$ is regularly varying with tail index ξ.*

This is a significant result linking the tail behaviour of the loss function to that of the underlying latent variable. One can understand the convergence (or not) towards a margin-maximiser in terms of the distributional properties of unobservable individual noise. The problem of convergence is thus connected to a separating margin-maximising hyperplane and heavy tails. Given Proposition 8, loss functions with different behaviours based on different underlying tail indices can now be produced.

2.6.3 Some examples

Some concrete examples of latent distributions that are regularly varying can be provided. For the sake of simplicity, only the class of elliptical distributions (see (Anderson, 2004) for a textbook treatment), which still covers the majority of known use cases, is considered. The evolution of the ratio $\ell(at)/\ell(t)$ for different types of distribution (with different tail behaviour) is illustrated in Figure 2.1; as per Figure 2.2, this is connected to tail behaviour of the underlying loss function and distribution. The case of the Gaussian and logistic distributions has already been tackled in Sections 2.2.1 and 2.2.2.

2.6.3.1 Cauchy distribution

The probability density function of a (standard) Cauchy distribution is given by

$$f_{\text{Cauchy}}(t) = \frac{1}{\pi(1+t^2)}$$

for all $t \in \mathbb{R}$. Its c.d.f. is $F_{\text{Cauchy}}(t) = \frac{1}{2} + \frac{1}{\pi}\arctan(t)$, hence

$$\ell_{\text{Cauchy}}(t) = -\log\left(\frac{1}{2} + \frac{1}{\pi}\arctan(t)\right). \tag{2.19}$$

From the fact that $\lim_{t\to+\infty} \frac{f_{\text{Cauchy}}(at)}{f_{\text{Cauchy}}(t)} = a^{-1}$, it is inferred that $\xi_{\text{Cauchy}} = 0$, i.e., the Cauchy distribution is *slowly* varying, and so is ℓ_{Cauchy}.

2.6.3.2 Student-*t* distribution

The Cauchy distribution is actually a particular case of a Student-*t* distribution, whose p.d.f. reads

$$f_{\text{Student}}(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

where $\nu \geq 1$ –the number of degrees of freedom– is a parameter governing the tail behaviour ($\nu = 1$ corresponds to the Cauchy case and $\nu = +\infty$ to the Gaussian one). Thus, the Student-*t* distribution has tail index $\xi_{\text{Student}} = \frac{\nu-1}{2}$, whence it has *regularly* varying tails for $\nu > 1$. It can also be deduced that $F_{\text{Student}}(t) = 1 - \frac{1}{2} I_{x(t)}\left(\frac{\nu}{2}, \frac{1}{2}\right)$, and $\ell_{\text{Student}}(t) = -\log\left(1 - \frac{1}{2} I_{x(t)}\left(\frac{\nu}{2}, \frac{1}{2}\right)\right)$, where $x(t) := \frac{\nu}{t^2+\nu}$ and I is the regularised incomplete beta function. Notice that the heaviness of the Student-t distribution's tails is an interesting feature, for example, by considering –as in (Shah et al., 2014)– Student-t processes instead of Gaussian ones.

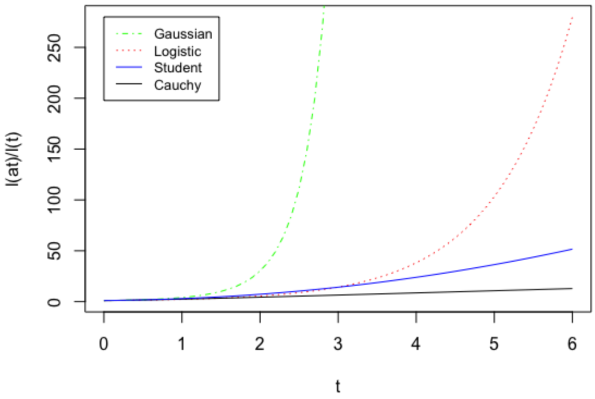

Figure 2.1: Evolution of the ratio $\ell(at)/\ell(t)$ as a function of t for loss functions associated respectively with the Gaussian, logistic, Student (with $\nu = 2$ degrees of freedom) and Cauchy distributions, and $a = 0.0001$.

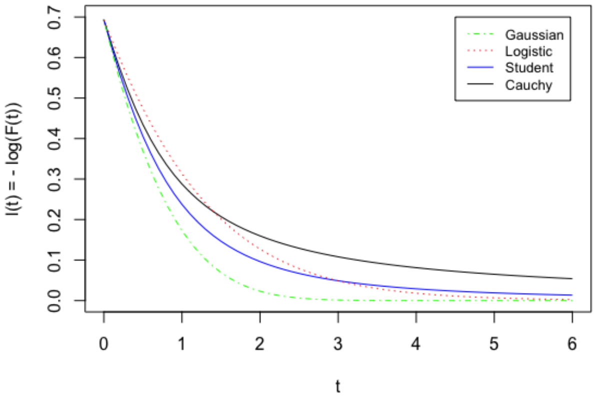

Figure 2.2: Tail behaviour of the respective loss functions $\ell(t) = -\log F(t)$, in the case of the Gaussian, logistic, Student (with $\nu = 2$ degrees of freedom) and Cauchy distributions.

It can be seen (in Figure 2.1) that the ratio statistic $t \mapsto \ell(at)/\ell(t)$ increases quickly in the Gaussian case, less quickly in the logistic case (while not visible on the plot, it converges in the Student and Cauchy examples). Heavy tails play a crucial role in robustness in statistics and machine learning (cf. (Hsu & Sabato, 2016; Huber & Ronchetti, 2009)) and show that loss functions may reveal different tail behaviours (cf. Figure 2.2) that have an impact on an algorithm's performance.

2.7 Numerical experiments

This Section is dedicated to experiments illustrating the impact of tail behaviour on classification tasks on real-life and surrogate datasets.

2.7.1 Symmetry gap and symmetrisation

The role of the underlying cumulative distribution function (c.d.f.) and its symmetry has been highlighted. The symmetry gap, $\Delta^{\mathrm{sym}}(t)$, can be introduced as follows:

$$\Delta^{\mathrm{sym}}(t) = 1 - F(-t) - F(t). \tag{2.20}$$

Suppose that for a given observation, the margin is worth m. The statistical *gain* is $F(m)$ (which, as per the maximum likelihood estimation interpretation from the previous section, we wish to maximise). On the other hand, if the margin were flipped, and an observation had a margin worth $-m$, then the incurred loss would be $1 - F(-m)$, in short:

$$\text{Symmetry gap} = \text{Loss of flipped margin} - \text{Gain of margin}.$$

This is, thus, an intrinsic measure of loss aversion for a given underlying distribution. Figure 2.3 shows the different symmetry gaps for the Gumbel and displaced exponential distributions. Note, in particular, the change in regime around -1 and 1 for the latter. This implies different trade-offs for different loss functions.

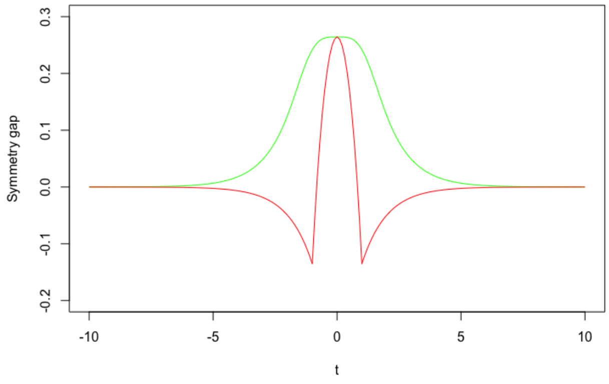

Figure 2.3: Symmetry gap associated with the exponential loss/Gumbel distribution (*green*) and the hinge loss/displaced exponential distribution (*red*)

2.7.2 Tail behaviour

As described earlier, in addition to symmetry, another essential element is the tail behaviour of the latent distribution. Recall that this is a key qualitative property of *heavier* tails of a distribution. The survival function $\overline{F}$ is considered. Under the assumption that ℓ is differentiable (or equivalently that F is differentiable, hence admits a probability density function f), it holds

$$\lim_{t\to+\infty} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = \lim_{t\to+\infty} \frac{\overline{F}(t(1-\varepsilon))}{\overline{F}(t)}. \tag{2.21}$$

In what follows, the tail criterion is said to be met for a given loss function ℓ if $\lim_{t\to+\infty} \frac{\ell(t(1-\varepsilon))}{\ell(t)} = +\infty$ for any $\varepsilon \in (0,1)$.

2.7.3 Comparison of loss functions

The probabilistic properties of margin loss functions can now be summarised in the following table[1].

Table 2.1: Main margin loss functions' key probabilistic properties

Loss function	**Definition**	**Latent distribution**	**Symmetry**	**Tail criterion**
Exponential	e^{-t}	Gumbel	No	Yes
Binomial Deviance	$\log(1+e^{-t})$	Logistic	Yes	Yes
Probit	$-\log(\Phi(t))$	Standard Gaussian	Yes	Yes
Hinge	$\max(0,1-t)$	Displaced Exponential	No	Yes
Huberised hinge	$-4t\mathbf{1}_{t<-1}+\max(0,1-t)^2\mathbf{1}_{t\geq-1}$	Exponential and Gaussian mixture	No	Yes
Negative Log Arctan	$-\log(\frac{1}{2}+\frac{1}{\pi}\arctan(t))$	Cauchy	Yes	No

2.7.4 Some numerical illustrations

To illustrate in practice the impact of tail behaviour on classification results via margin-maximisation, two datasets are considered, namely a real-life dataset and a surrogate one. Linear classifications are run with different loss functions, including loss functions corresponding to regularly varying distributions. Results across loss functions are necessarily close since each setting applies different margin loss functions to the same linear model. However, they display different behaviour and show the good performance of the negative log arctan loss function. The key point is not so much the numerical results linked to a particular loss function, as choosing more robust loss functions may be helpful.

2.7.4.1 South African heart disease dataset

The South African Heart Disease dataset[2] is a retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. It contains 463 observations of eight input features and one binary response. The features are systolic blood pressure, cumulative tobacco (kg), low-density lipoprotein cholesterol, adiposity, family history of heart disease (Present, Absent), obesity, current alcohol consumption, age at onset, while the binary *response variable* is coronary heart disease (± 1).

[1] The symbol $\mathbf{1}_{t\in A}$ is worth 1 if $t \in A$ and 0 otherwise.

[2] The South African Heart Disease dataset is available at http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data

Results can now be presented. The results are very close across loss functions. In what follows, heavier tails are introduced to see if the choice of loss function starts impacting cross-validated performance.

Table 2.2: Accuracy, false positive and negative rates for different loss functions on the South African heart dataset.

Loss function	**Accuracy**	**False Positive Rate**	**False Negative Rate**
Exponential	73%	47%	16%
Binomial Deviance	73%	48%	15%
Probit	73%	48%	15%
Hinge	73%	47%	16%
Huberised hinge	74%	47%	15%
Negative Log Arctan	74%	48%	15%

2.7.4.2 Surrogate dataset

A dataset of five hundred observations made up of two features and one binary response variable was also generated. For each observation, the features and response variable are independently generated according to the following procedure:

$$\begin{aligned} x_1 &\sim \text{Bernoulli}(1/2) \\ x_2 &\sim \text{Normal}(0,1) \\ y &= \text{sign}(\alpha + \beta_1 x_1 + \beta_2 x_2 + \sigma\varepsilon), \end{aligned}$$

where $\varepsilon \sim$ Student(3), $\alpha = 0.1$. $\beta_1 = 1$, $\beta_2 = -2$ and $\sigma = 2$.

Table 2.3: Accuracy, false positive and negative rates for different loss functions on a synthetic dataset.

Loss function	**Accuracy**	**False Positive Rate**	**False Negative Rate**
Exponential	75.4%	22.4%	27.2%
Binomial Deviance	74.8%	21.3%	29.7%
Probit	75.0%	21.6%	28.9%
Hinge	75.4%	21.6%	28.0%
Huberised hinge	75.0%	21.3%	29.3%
Negative Log Arctan	76.2%	18.3%	30.2%

In addition to these classification results, the difference between the real weights specified above and the ones derived from minimising each loss function is also measured. The weights are renormalised so that each weight vector has a unit norm. Performance is varied across estimators, and the Negative Log Arctan has the best results on this surrogate dataset (albeit not by much).

2.8 Conclusions

The primary focus of the work (Freund & Schapire, 1997; Friedman et al., 2000; Rosset et al., 2003, 2004; Schapire et al., 1997) that spurred the present chapter was the relationship between support

Table 2.4: Mean absolute and root mean squared errors between the true weight vector and the weight vector estimated via different loss functions on the surrogate dataset.

Loss function	**Mean Absolute Error (MAE)**	**Root Mean Squared Error (RMSE)**
Exponential	9.45%	7.11%
Binomial Deviance	5.76%	4.29%
Probit	7.72%	5.78%
Hinge	16.36%	12.45%
Huberised hinge	8.31%	6.23%
Negative Log Arctan	2.12%	1.60%

vector machines and regularisation, and their respective benefits and drawbacks. To some extent, the limiting criterion in Eq. (2.3) is a necessary condition too, but further research is warranted to weaken assumptions.

More importantly, the margin-maximisation property of classifiers (such as support vector machines (Vapnik, 1998)) was considered as a benchmark for classification tasks and the properties of loss functions that do not lead to the convergence of the normalised regularised estimator to a margin-maximising classifier were determined.

Surprisingly, this is the case (under mild assumptions) if and only if the loss function is regularly varying, which is equivalent to the underlying latent distribution having heavy tails. The results presented here, while giving a precise quantitative characterisation, are more qualitative in nature, in the sense that they highlight two possible regimes in terms of the behaviour of the normalised and regularised classifier. While usual loss functions that as the exponential, hinge, Probit or logistic loss have similar behaviour (in terms of convergence in the separable case), heavy-tailed loss functions have fundamentally different properties.

From a more practical perspective, it also shows that relying on usual loss functions assumes that there are no heavy tails. This finding is not limited to purely linear or dictionary learning models but extends to all methods using a margin-dependent loss function (i.e., the dictionary $\mathscr{H}$ need not be fixed). Heavy tails are a growing and exciting part of the recent machine learning literature (Hsu & Sabato, 2016; Lugosi & Mendelson, 2019) and distribution estimation (Ben-Hamou et al., 2017), and open exciting perspectives for real-life data as the presence of heavy tails is well-documented (Taleb, 2020). Some questions remain around the application of these insights to multi-class classification and the impact of regularly varying loss functions in other settings, such as deep neural networks or Gaussian processes for classification.

Chapter 3

Kernel Factorisation Machines

This chapter is the result of a paper co-authored with Islam Utyagulov. F.B.G. conceived of the presented ideas developed the theoretical aspects, designed the experiments, and wrote the manuscript. I.U. contributed to the code base and ran the numerical experiments. Both authors discussed the results and commented on the manuscript.

Research objectives

Recommender systems are a type of information filtering system that is designed to predict and suggest items of interest to users, such as products, services, or content. They are widely used in many applications, including e-commerce, social media, entertainment, and healthcare, and have become an essential tool for enabling personalised and targeted recommendations to users.

The goal of a recommender system is to provide accurate and relevant recommendations to users based on their preferences, interests, and behaviour. To achieve this goal, recommender systems typically use various techniques such as collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering is based on the idea that users who have similar preferences in the past are likely to have similar preferences in the future. Content-based filtering, on the other hand, uses the characteristics of the items themselves to generate recommendations. Hybrid approaches combine both collaborative and content-based filtering to provide more accurate and diverse recommendations. Recommender systems have become increasingly important in recent years due to the growth of online data and the increasing demand for personalised and targeted recommendations. They have also become more complex and sophisticated, incorporating techniques such as deep learning, reinforcement learning, and explainable AI to improve the accuracy and interpretability of the recommendations.

However, despite the many advances in recommender systems, there are still many challenges and open research questions in this area, such as how to handle cold start problems, how to ensure fairness and diversity in the recommendations, and how to provide transparent and trustworthy recommendations to users. Such algorithms seem sometimes to be very task specific (e.g., perform well on a particular dataset but not another one) and consume significant amounts of energy to be trained. The challenge tackled in this chapter is thus twofold: first, propose a model that is flexible enough to tackle the small- and big-data setups; second, ensure that it is not (too) sensitive to the curse of dimensionality and remains parsimonious.

3.1 Introduction

This chapter explores a generalisation of factorisation machines via kernels, termed Kernel Factorisation Machines ("KFM"). It is well-known that functions in reproducing kernel Hilbert spaces can be understood as a linear combination of features in very high-dimensional (or infinite-dimensional) spaces while being computed in a finite-dimensional space, thanks to the representer theorem (Hastie et al., 2009; Scholkopf & Smola, 2001). Furthermore, it has been shown recently that the dot product operation was a key component behind the success of several recommender systems (Rendle et al., 2020), while the recent literature has been preoccupied with enriching factorisation machines. Thus, a framework is needed to interpolate between factorisation machines that tend to outperform other techniques on sparse datasets and more advanced models that perform well on large and dense datasets.

One of the drawbacks of kernel methods is their high dimensionality when the number of observations is large (say at least in the thousands), which is typical of recommender systems. It is thus extremely important to be able to reduce the dimensionality, which is accomplished in two different ways in this chapter: first, by finding a representation of the input features in a lower-dimensional space, and second, by considering inducing points, i.e., surrogate inputs that are optimised upon training to avoid building (kernel) interactions between each pair of observations in the dataset. In short, a method that adapts kernels to the set-up of high-dimensional and potentially sparse datasets is proposed. To illustrate this approach, it is tested on four well-known datasets, and its results are benchmarked against most available models. While comparisons are difficult and should be interpreted carefully, KFM can perform well and obtains the best performance overall. The proposed methodology is not limited to recommender systems and can be applied to other settings, as illustrated by a heart disease classification task.

Embedding-based models have efficiently tackled collaborative filtering since the Netflix challenge (and matrix factorisation techniques, see (Udell et al., 2016) for a textbook treatment of related techniques) and the advent of factorisation machines ("FM") (Rendle, 2010). In particular, such models combine some embeddings, usually customer and item latent variables, to derive a likelihood for a customer to adopt or review an item. Matrix factorisation, for instance, models this interaction by a dot product. Since then, much effort has been spent on generalising this approach to more complex functions and encompassing more information with additional embeddings or context. Beyond factorisation machines, many models have been put forward, such as field-aware FMs (Juan et al., 2016), higher-order FMs (Blondel et al., 2016), factorisation-supported neural networks (Zhang et al., 2016), wide and deep (Cheng et al., 2016), etc., attention FM (Xiao et al., 2017), neural FM (He & Chua, 2017), neural collaborative filtering (He et al., 2017), field-aware neural FMs (Zhang et al., 2019), product neural network (Qu et al., 2018), deep cross network (Wang et al., 2017), deep FM (Guo et al., 2017), extra-deep FM (Lian et al., 2018), automatic feature interaction model (Song et al., 2019) and adaptive factorisation network model (Cheng et al., 2020) amongst others, generally by relaxing assumptions made in the seminal FM paper (Rendle, 2010) and considering more flexible functions (e.g., by introducing deep neural networks).

However, as pointed out recently in (Rendle et al., 2020), the dot product operation in such recommender systems is essential. It explains the very good performance of matrix factorisation and factorisation machines. The intuition behind the approach proposed in this chapter is thus to preserve this operation but apply it in a more "complex" feature space, namely that defined via a kernel. Kernel methods ((Scholkopf & Smola, 2001; Shawe-Taylor & Cristianini, 2004)) have been widely used in machine learning, especially support vector machines, but less so in recommender systems (except for (Blondel et al., 2016)). One of the issues when using kernel methods is their relatively poor performance on large datasets due to the curse of dimensionality. To tackle this challenge, embeddings are created on the inputs. Instead of considering all observations as possible support vectors, inducing points are introduced in the Gaussian process literature (Duvenaud, 2014; Rasmussen & Williams, 2005).

Contributions. The main aspects of the KFM approach are as follows: first, a generalised representer theorem is introduced to handle multiple reproducing kernel Hilbert spaces; second, a method to perform dimensionality reduction via embeddings and inducing points is described; third, it is established that kernel design can be performed at the same time. By considering multiple tests, it is shown that KFMs perform very well in both small- and big-data regimes.

3.2 Background

In this Section, the building blocks of KFM are introduced, and the focus of the discussion is on kernels; for a textbook treatment of the latter, the reader is referred to (Scholkopf & Smola, 2001; Shawe-Taylor & Cristianini, 2004) and Chapter 5 in (Hastie et al., 2009).

3.2.1 Overview of kernels

Consider a positive definite kernel K and the reproducing kernel Hilbert ("RKHS") space $\mathscr{H}_K$ linked to K. One can now suppose –for instance, thanks to Mercer's theorem (Scholkopf & Smola, 2001)–, that K has an eigenvalue expansion given by $K(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{+\infty} \gamma_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$, where $\gamma_i \geq 0$ and $\sum_{i=1}^{+\infty} \gamma_i^2 < +\infty$ and the ϕ_i's are the related eigenfunctions. Then, elements of $\mathscr{H}_K$ have the following expression in terms of this eigenbasis, $f(\mathbf{x}) = \sum_{i=1}^{+\infty} c_i \phi_i(\mathbf{x})$, where the coefficients c_i verify that $\|f\|_K^2 := \sum_{i=1}^{+\infty} c_i^2/\gamma_i$

is finite ($\|\cdot\|_K$ is the norm induced by K). Turn to the empirical risk minimisation problem:

$$\min_{f\in\mathscr{H}_K}\sum_{i=1}^{N}\ell_i(y_i,f(\mathbf{x}_i))+\mathscr{P}(\|f\|_K^2), \tag{3.1}$$

where ℓ_i is the loss associated with a regression or classification task and can be different for each observation i and $\mathscr{P}$ is a strictly increasing real-valued penalty function (which is often taken to be $\mathscr{P}(r)=\lambda r$, where $\lambda, r\geq 0$).

The key result on RKHS is the representer theorem, according to which the solution to Eq. 3.1 is finite-dimensional, i.e., f can be expressed as $f(\mathbf{x})=\sum_{i=1}^{N}\alpha_i K(\mathbf{x},\mathbf{x}_i)$ and boils down to

$$\min_{\alpha\in\mathbb{R}^N}\sum_{i=1}^{N}\ell_i\left(y_i,\sum_{j=1}^{N}\alpha_j K(\mathbf{x},\mathbf{x}_j)\right)+\mathscr{P}\left(\alpha^T\mathbf{K}\alpha\right), \tag{3.2}$$

with $\mathbf{K}$ the kernel Gram matrix such that $\mathbf{K}_{i,j}=K(\mathbf{x}_i,\mathbf{x}_j)$. The shorthand $\sum_{j=1}^{N}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)=\alpha^T\mathbf{k}_i$ is also used with a slight abuse of notation.

3.2.2 A kernel viewpoint on FMs

In the case of a logistic regression (or a linear model), the function f is linear; hence f can be understood as belonging to an RKHS with a linear kernel, i.e., $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$. Similarly, in FMs, f is given by $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+\sum_{r=1}^{d}\left(\mathbf{v}_r^T\mathbf{x}\right)^2$. Importantly, the latter can be immediately rewritten as

$$\begin{aligned} f(\mathbf{x}) &= h\left(\mathbf{w}^T\mathbf{x},\mathbf{v}_1^T\mathbf{x},\cdots,\mathbf{v}_d^T\mathbf{x}\right)\\ &= h\left(f_0(\mathbf{x}),f_1(\mathbf{x}),\cdots,f_d(\mathbf{x})\right), \end{aligned}$$

where each coordinate function, f_i, for $i=0,\cdots,d$ belongs to the RKHS associated with the linear kernel on the original feature space. A generalised framework is introduced in which the f_i's can belong to arbitrary (and different) RKHSs.

3.3 A generalised representer theorem

This Section extends the representer theorem to the set-up with multiple functions. The regularised empirical risk function can be defined as:

$$\sum_{i=1}^{N}\ell_i\left(y_i,h\left(f_1(\mathbf{x}_i),\cdots,f_d(\mathbf{x}_i)\right)\right)+\mathscr{P}\left(\|f\|_{K_1}^2,\cdots,\|f\|_{K_d}^2\right), \tag{3.3}$$

where $f_i\in\mathscr{H}_{K_i}$ for $i=1,\cdots,d$ and $\mathscr{P}$ is strictly increasing in each of its d components. Then, the following result holds:

Proposition 10. *The minimisation of the penalised empirical loss function with respect to each RKHS* *is equivalent to solving a finite-dimensional problem:*

$$\min_{f_1 \in \mathscr{H}_{K_1}, \cdots, f_d \in \mathscr{H}_{K_d}} \left\{ \sum_{i=1}^{N} \ell_i \left(y_i, h(f_1(\mathbf{x}_i), \cdots, f_d(\mathbf{x}_i))\right) + \mathscr{P}\left(\|f\|^2_{K_1}, \cdots, \|f\|^2_{K_d}\right) \right\}$$
$$= \min_{\alpha_1, \cdots, \alpha_d \in \mathbb{R}^N} \left\{ \sum_{i=1}^{N} \ell_i \left(y_i, h\left(\alpha_1^T \mathbf{k}_{1,i}, \cdots, \alpha_d^T \mathbf{k}_{d,i}\right)\right) + \mathscr{P}\left(\alpha_1^T \mathbf{K}_1 \alpha_1, \cdots, \alpha_d^T \mathbf{K}_d \alpha_d\right) \right\}, \quad (3.4)$$

The proof is omitted as it is a straightforward extension of (Scholkopf & Smola, 2001) (see also Exercise 5.15 in (Hastie et al., 2009)). This result can immediately be extended to more general error functions, as in (Scholkopf & Smola, 2001).

3.4 Finding a smaller space

As noted in (Rendle, 2010) and in the SVM literature (Scholkopf & Smola, 2001), kernel methods suffer from the curse of dimensionality as N grows. An important task is thus to address the latter.

3.4.1 Truncating the number of features

One of the issues with using RKHS is that, while the number of parameters to estimate is finite, it scales linearly in the dimension of the available sample. This is particularly problematic for recommender systems as sample sizes can be large. A possible way of dealing with this challenge is to truncate the kernel's feature expansion and choose a finite number of eigenfunctions, S, rather than the (usually) infinite expansion:

$$f(\mathbf{x}) = \sum_{i=1}^{S} c_i \phi_i(\mathbf{x}), \quad (3.5)$$

which is equivalent to truncating the kernel itself and introducing K' as $K'(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{S} \gamma_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$. Furthermore, by introducing the space $\mathscr{H}' := \left\{\sum_{i=1}^{S} c_i \phi_i(\cdot)\right\} \subset \mathscr{H}$, the natural inner product $\langle f, g \rangle_{\mathscr{H}'} = \sum_{i=1}^{S} \frac{c_{f,i} c_{g,i}}{\gamma_i}$ can be used.

3.4.2 Choosing a small set of support vectors

Suppose that for $s = 1, \cdots, S$, there exists a feature vector $\mathbf{z}_s \in \mathscr{X}$ and that the space $\mathscr{H}''$ of linear combinations of the basis functions $K(\cdot, \mathbf{z}_s)$ is considered

$$\mathscr{H}'' := \left\{ \sum_{s=1}^{S} \alpha_s K(\cdot, \mathbf{z}_s) \right\}, \quad (3.6)$$

$$f(\mathbf{x}) = \sum_{s=1}^{S} \alpha_s K(\mathbf{x}, \mathbf{z}_s) \quad (3.7)$$

In particular, if f belongs to $\mathscr{H}''$ (with corresponding weight vector α_f then it also belongs to $\mathscr{H}$, from which important properties are inherited. For instance, by using the scalar product in $\mathscr{H}$, one sees that for two functions $f, g \in \mathscr{H}''$, $\langle f, g \rangle_{\mathscr{H}''} = \langle f, g \rangle_{\mathscr{H}} = \alpha_f^T \mathbf{K}_Z \alpha_g$. Similarly, $\|f\|^2_{\mathscr{H}''} = \|f\|^2_K = \alpha_f^T \mathbf{K}_Z \alpha_f$. In addition, for $f \in \mathscr{H}'$, one recovers the equality $\langle K(\cdot, \mathbf{x}_i), f(\cdot) \rangle_{\mathscr{H}} = f(\mathbf{x}_i)$. In short,

$\mathscr{H}''$ inherits many of the important properties of $\mathscr{H}$. To some extent, this approach can be seen as specifying a fixed number of support vectors $\mathbf{z}_s$.

3.4.3 Comparing both approaches

Both $\mathscr{H}'$ and $\mathscr{H}''$ are subspaces of $\mathscr{H}$ and, in both cases, there are S (supposed to be such that $S \ll N$) parameters to fit: either $c_1, \cdots c_S$ in $\mathscr{H}'$ or $\alpha_1, \cdots, \alpha_S$ in $\mathscr{H}''$. It is well-known (Hastie et al., 2009; Scholkopf & Smola, 2001) that approximating a function in $\mathscr{H}$ with its first S elements is optimal in the sense that it minimises the approximation error with respect to the $\|\cdot\|_K$-norm. On the other hand, for any function $f \in \mathscr{H}''$, the following expansion holds:

$$f(\mathbf{x}) = \sum_{s=1}^{S} \alpha_s K(\mathbf{x}, \mathbf{z}_s) = \sum_{i=1}^{+\infty} \left[\gamma_i \sum_{s=1}^{S} \alpha_s \phi_i(\mathbf{z}_s) \right] \phi_i(\mathbf{x}). \tag{3.8}$$

In other words, $c_i = \gamma_i \sum_{s=1}^{S} \alpha_s \phi_i(\mathbf{z}_s)$. Functions in $\mathscr{H}''$ thus have a (generally) non-zero loading on *all* eigenfunctions ϕ_i, whereas functions in $\mathscr{H}'$ are constrained to have a zero loading on all eigenfunctions ϕ_i for $i = S+1, \cdots$. Thus, $\mathscr{H}''$ retains some expressiveness across all eigenfunctions. This option is the one chosen in KFMs.

Remark 7. Other methods exist that produce a finite-dimensional feature approximation apart from truncation. Among the classic methods are the random Fourier features (Rahimi & Recht, 2007) and Nystrom approximations (Zhang et al., 2008) (which can be seen as a specific way of choosing a subset of points). One should compare these, both theoretically and empirically, against the proposed method, which is left for future research.

3.5 Kernel factorisation machines

The essence of kernel factorisation machines is to retain the flexible formulation (in Eq. 3.3), while avoiding the curse of dimensionality. To do so, two techniques are relied upon: kernel design and inducing points.

3.5.1 Kernel design

So far, it has been assumed that the kernels in use were given (for instance, K is known or its eigenfunctions ϕ_i are). One successful avenue, particularly explored in Gaussian processes, is the parametrisation of kernel functions during training, rather than their *ex-ante* specification. This topic is not discussed in its generality (Chapters 2 and 3 in (Duvenaud, 2014) are particularly relevant), but three ideas relevant to KFMs are introduced. From now on, it is supposed that the input feature space $\mathscr{X}$ is in $\mathbb{R}^p$.

First, if K is a (valid) kernel on $\mathbb{R}^q \times \mathbb{R}^q$, then $K' : (\mathbf{x}, \mathbf{y}) \mapsto K(\mathbf{V}\mathbf{x}, \mathbf{V}\mathbf{y})$, where $\mathbf{V} \in \mathbb{R}^{q \times p}$, is a valid kernel on $\mathbb{R}^p \times \mathbb{R}^p$. The matrix $\mathbf{V}$ can be understood as a linear embedding and performs dimensionality reduction if $q < p$. In the particular case where K is the dot product, $K(\mathbf{V}\mathbf{x}, \mathbf{V}\mathbf{y}) = \mathbf{x}^T \mathbf{W} \mathbf{y}$, where the matrix $\mathbf{W} = \mathbf{V}^T \mathbf{V}$ is positive semi-definite but low rank, which is similar to FMs.

Second, kernels are generally parametric functions. For example, a polynomial kernel can be expressed as $K(\mathbf{x},\mathbf{y}) = (c+\mathbf{x}^T\mathbf{y})^{\mathrm{deg}}$, where $c \geq 0$ and $\mathrm{deg} = 1,2,\cdots$. While one may choose the dimension d to be fixed, the bias term c can be set during training. Similarly, the Automatic Relevance Determination ("ARD") Squared Exponential kernel is defined as

$$K(\mathbf{x},\mathbf{y}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{m=1}^{p}\frac{(\mathbf{x}_m-\mathbf{y}_m)^2}{\ell_m^2}\right), \tag{3.9}$$

with $\sigma > 0$, leaving $p+1$ parameters to be calibrated.

Third, as in the linear embedding case, one can map input features to an alternative space and then apply a kernel on this space. This is generally achieved by defining the kernel as $K(\mathbf{x},\mathbf{y}) = h_\theta(\mathbf{x})^T h_\theta(\mathbf{y})$, where h_θ is a function from $\mathbb{R}^p$ to $\mathbb{R}^q$, parametrised by a parameter vector θ, that can be determined via a neural network.

Remark 8. By introducing the space $\mathscr{H}''$, the vectors $\mathbf{z}_s$, for $s = 1,\cdots,S$ have not been specified and can thus be seen as free parameters. One could randomly select S available input features $\mathbf{x}_i$ (i.e., choose them in the training dataset), but they need not be actual data points. The terminology of Gaussian processes is used, where such points are referred to as *inducing* points in the sense that they help "summarise" the data and are fitted upon training. This connection to Gaussian processes suggests that another avenue could be explored via deep Gaussian processes (Damianou & Lawrence, 2013). Indeed, deep Gaussian processes can "compose" kernels and handle large amounts of data; however, it is unclear whether such "depth" is needed given that factorisation machines and simple KFMs perform well.

3.5.2 A KFM toy example

A simple example of a function we can fit via KFM based on the linear and squared exponential kernels is as follows:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \sum_{r=1}^{d}(\mathbf{v}_r^T\mathbf{x})^2 + \sum_{s=1}^{S}\alpha_s \exp(\|\mathbf{V}\mathbf{z}_s - \mathbf{V}\mathbf{x}\|^2), \tag{3.10}$$

where $\mathbf{V}$ is a $\mathbb{R}^{q\times p}$ matrix. The first term in f's expression can be seen as a linear model, the second term is a low-rank quadratic form to capture feature interactions (and corresponds to a factorisation machine), whereas the third term accounts for higher-order terms. Setting all α_s's to 0 yields the usual FM, whereas Higher-Order FMs (Blondel et al., 2016) can be expressed in that way after replacing the squared exponential kernel with an ANOVA kernel.

Note that, since the inducing points $\mathbf{z}_1,\cdots,\mathbf{z}_S$ have to be calibrated as well as the matrix $\mathbf{V}$, one can optimise the d-dimensional parameter $\mathbf{z}'_s = \mathbf{V}\mathbf{z}_s$ directly and rewrite f as

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \sum_{r=1}^{d}(\mathbf{v}_r^T\mathbf{x})^2 + \sum_{s=1}^{S}\alpha_s \exp(\|\mathbf{z}'_s - \mathbf{V}\mathbf{x}\|^2). \tag{3.11}$$

Practical considerations When estimating these parameters by minimising the empirical loss function, one also has to include the penalty term $\alpha^T \mathbf{K}_Z \alpha$, where $\mathbf{K}_Z$ is the kernel Gram matrix ($\mathbf{K}_{u,v} = \exp(\|\mathbf{z}'_u - \mathbf{z}'_v\|^2)$ for $u, v = 1, \cdots, S$), which thus also depends on the $\mathbf{z}'_s$'s. In practice, however, no improvement has been noticed from taking this dependence into account in the penalty term so that one can choose the usual Tikhonov regularisation $\alpha^T \alpha$ instead. If one is unsure of the number of inducing points to pick, one can set a high cardinal S but add an L^1 penalty term on the vector α to impose some sparsity. Similarly, if one wishes to select features and not just reduce dimensionality, one can also set an L^1 penalty on the $\mathbf{v}_r$'s and $\mathbf{V}$.

3.5.3 Possible architectures

Before moving to the empirical part of the paper (see Section 3.6, two different architectures are applied, respectively, on the South-African Heart Disease dataset (which is small) and on four recommender system benchmarking datasets (which are large). Due to the smaller size of the first dataset, a straightforward architecture was employed (see Figure 3.1: the output is the sum of two components: 1) a linear embedding, and 2) the output of squared exponential kernel, which is applied to a linear projection. This corresponds to the toy KFM in Eq. 3.11 where the quadratic terms have been set to 0 (i.e., $\mathbf{v}_r = 0$, for $r = 1, \cdots, d$). The KFM architecture used for recommender systems (cf. Figure 3.2) consists of the following building blocks: 1) a linear embedding and a weighted combination thereof, 2) second-order interaction terms obtained as in FMs, and 3) the output of polynomial kernel, which is applied to a shallow neural network (thus creating a very simple neural

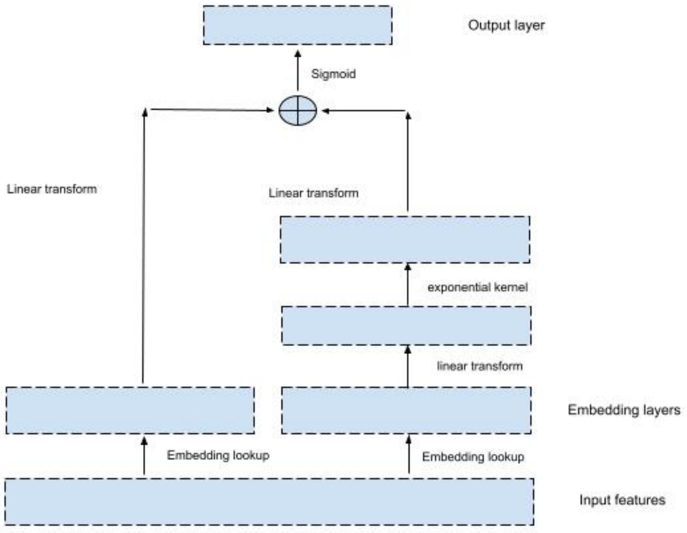

Figure 3.1: Schematic overview of the KFM architecture used on the South African Disease dataset.

network kernel).

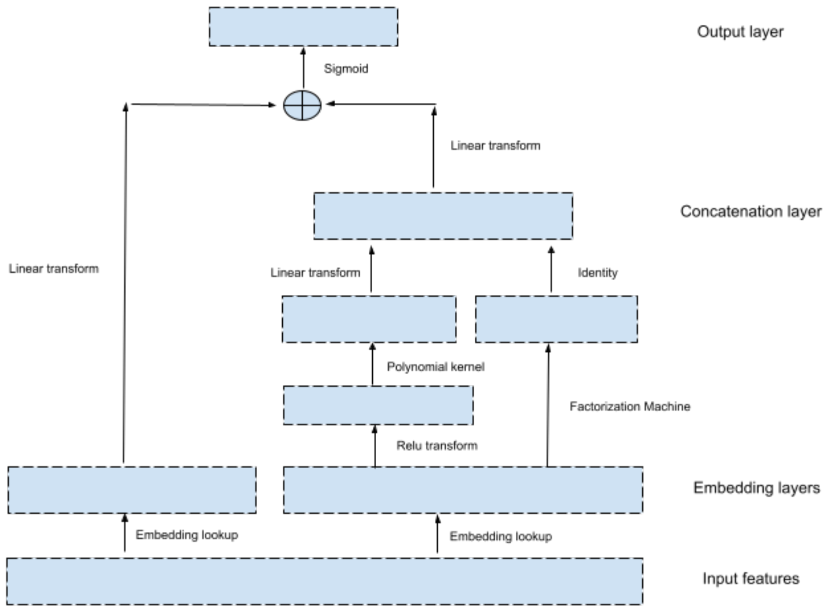

Figure 3.2: Schematic overview of the KFM architecture used on the MovieLens, Avazu and Criteo datasets.

3.6 Empirical tests and results

To investigate the empirical properties of KFM, two different set-ups are studied: a small dataset (the South African heart disease dataset) and typical benchmark datasets used in the recommender system literature. The objective is to test KFM under different regimes of data and determine its reliability.

3.6.1 KFM for supervised learning: the South African heart disease dataset

As mentioned earlier, models used in recommender systems specifically try to model interactions between embeddings for users and items. Still, they can more broadly tackle other functions, such as a traditional classification task with a smaller set of features and observations[1]. KFM is compared with several other well-known techniques, such as logistic regression (with and without regularisation), gradient-boosted classifier, random forest, support vector machine (with linear, polynomial (of third degree) and Gaussian kernels respectively) and, finally, factorisation machine (with an embedding dimension of 3).

Results As can be seen in Table 3.1, KFM, FM and regularised logistic regression have similar performances and seem to outperform other methods, such as SVM. This agrees with findings in

[1] The South African Heart Disease dataset was described in the previous chapter. As a reminder, it is a retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. It is a "small" dataset as it contains 463 observations of eight input features and one binary response. The features are systolic blood pressure, cumulative tobacco (kg), low-density lipoprotein cholesterol, adiposity, family history of heart disease (present, absent), obesity, current alcohol consumption, and age at onset, while the response variable is the presence (or not) of coronary heart disease.

Table 3.1: Performance comparison of different models on the South African Heart Disease dataset. The average and standard deviation of ROC AUC are computed on five randomly chosen test sets.

Model	Average AUC	Stdev AUC
Logistic Regression	76.85%	3.12%
Regularised LR	77.10%	3.03%
Gradient Boosting Classifier	68.62%	3.89%
Random Forest	71.30%	4.05%
Linear SVM	76.65%	3.16%
Kernel SVM	73.25%	2.88%
Polynomial SVM	74.16%	4.30%
FM	77.66%	2.18%
KFM	77.30%	2.52%

(Rendle, 2010) that examined the comparative performance of FM and SVM and reached the same conclusion on multiple datasets. This phenomenon was investigated by varying the number of inducing points. The performance on the South African Heart Disease dataset was found to be best when the number of inducing points was less than five, cf. Figure 3.3, which can be understood as a sparsity requirement.

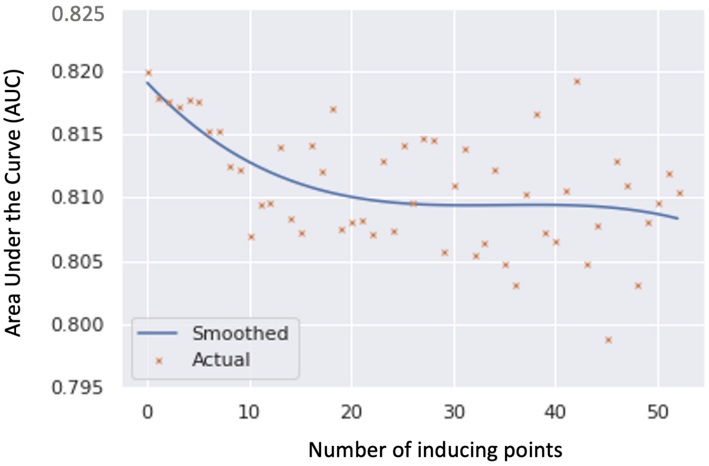

Figure 3.3: Sensitivity of the KFM's test area-under-the-curve (AUC) to the number of inducing points on the South-African Heart Disease dataset.

3.6.2 KFM for recommender systems: MovieLens, Avazu and Criteo datasets

To perform proper testing and benchmarking of KFMs, KFMs and other algorithms were run on well-known standard datasets: the *MovieLens* 1 million and 20 million rating datasets[2], the *Avazu* click-through-rate dataset[3] and the *Criteo* click-through-rate dataset[4]. Benchmarking recommender systems is notoriously difficult and test results have often lacked reproducibility, as demonstrated in (Dacrema et al., 2021) and (Rendle et al., 2020). To do so, the Pytorch FM library[5] was employed,

[2] https://grouplens.org/datasets/movielens/
[3] https://www.kaggle.com/c/avazu-ctr-prediction
[4] https://www.kaggle.com/c/criteo-display-ad-challenge
[5] https://rixwew.github.io/pytorch-fm/

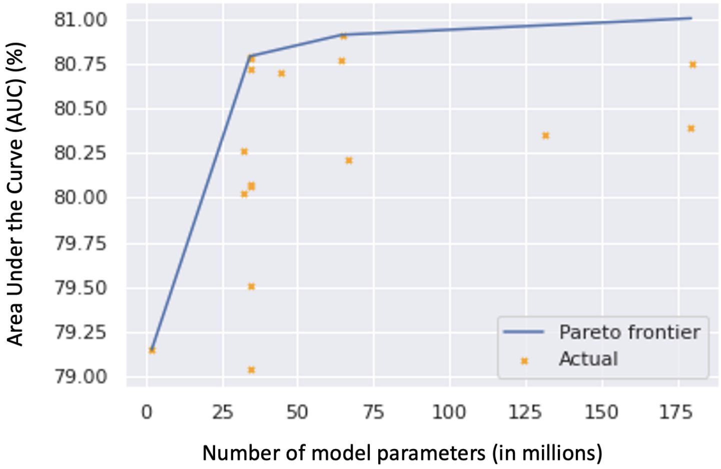

Figure 3.4: Test area-under-the-curve (AUC) (in %) vs utilised number of parameters in different architectures. Datapoints are taken from benchmarking on the Avazu dataset.

Table 3.2: Performance statistics across various methods on four different data sets. Area-under-the-curve (AUC) and log loss are reported on the test set. The time in minutes shows the average time needed to complete 1 epoch of training. The number of parameters corresponds to the number of trainable parameters of each model.

		Movielens 1M				Movielens 20M				Avazu				Criteo			
Model	Reference	AUC	logloss	time, min	#params	AUC	logloss	time, min	#params	AUC	logloss	time, min	#params	AUC	logloss	time, min	#params
LR		78.97%	0.544	0.08	9,993	79.59%	0.529	1.2	269,756	75.61%	0.395	3.0	2,018,013	79.15%	0.458	3.4	1,086,811
FM	(Rendle, 2010)	81.33%	0.517	0.07	169,865	83.70%	0.481	1.0	4,585,836	77.08%	0.384	3.4	34,306,205	79.04%	0.461	4.1	18,475,771
FFM	(Juan et al., 2016)	80.15%	0.529	0.07	89,929	82.64%	0.494	1.3	2,427,796	77.91%	0.378	17.5	179,603,069	80.39%	0.446	42.8	170,629,171
HOFM	(Blondel et al., 2016)	81.22%	0.518	0.09	329,737	83.69%	0.481	1.9	8,901,916	77.65%	0.382	7.7	66,594,397	80.21%	0.450	8.3	35,864,731
AFI	(Song et al., 2019)	76.74%	0.860	0.10	397,691	82.37%	0.512	2.4	4,813,662	78.16%	0.376	5.4	34,663,311	80.78%	0.440	11.5	18,942,765
AFM	(Xiao et al., 2017)	79.12%	0.546	0.08	170,171	81.66%	0.504	1.5	4,586,142	77.28%	0.383	5.6	34,306,511	80.06%	0.449	10.8	18,476,077
AFN	(Cheng et al., 2020)	77.25%	0.622	0.08	10,096,866	82.89%	0.493	4.8	14,512,837	77.73%	0.381	7.2	44,263,206	80.70%	0.445	12.1	28,458,272
DCN	(Wang et al., 2017)	79.86%	0.542	0.10	160,977	81.72%	0.508	1.7	4,317,185	77.98%	0.380	4.2	32,296,657	80.26%	0.450	4.2	17,403,681
DFM	(Guo et al., 2017)	79.64%	0.539	0.10	170,746	83.46%	0.483	1.1	4,586,717	78.08%	0.379	4.3	34,312,206	79.51%	0.457	4.0	18,486,124
FNFM	(Zhang et al., 2019)	79.80%	0.535	0.08	90,450	81.59%	0.510	1.6	2,428,317	78.63%	0.376	25.6	179,664,310	80.75%	0.445	43.7	170,825,052
FNN	(Zhang et al., 2016)	79.90%	0.539	0.08	160,753	81.79%	0.503	1.0	4,316,961	77.70%	0.382	4.0	32,294,193	80.02%	0.456	4.0	17,399,313
NCF	(He et al., 2017)	79.93%	0.538	0.08	160,769	83.52%	0.489	1.5	4,316,977								
NFM	(He & Chua, 2017)	75.96%	0.783	0.10	653,962	81.92%	0.507	1.3	17,538,557	77.65%	0.381	9.6	131,175,262	80.35%	0.448	5.4	70,647,132
IPNN	(Qu et al., 2018)	76.99%	0.785	0.08	320,353	83.08%	0.500	1.4	8,632,769	78.04%	0.382	3.5	64,585,793	80.77%	0.446	4.2	34,799,841
OPNN	(Qu et al., 2018)	77.31%	0.700	0.08	320,609	83.11%	0.498	1.5	8,633,025	78.23%	0.379	4.6	64,644,929	80.91%	0.444	25.4	34,989,537
WD	(Cheng et al., 2016)	79.51%	0.544	0.10	170,746	81.52%	0.506	1.1	4,586,717	77.38%	0.384	3.5	34,312,206	80.07%	0.453	4.0	18,486,124
XDFM	(Lian et al., 2018)	77.20%	0.617	0.08	171,387	82.20%	0.508	2.9	4,587,358	77.96%	0.379	6.0	34,325,647	80.79%	0.442	6.8	18,520,509
KFM		80.75%	0.528	0.07	170,207	83.75%	0.448	1.4	4,586,178	78.16%	0.379	4.9	34,309,748	80.72%	0.444	5.2	18,482,034

which has standardised factorisation machine-type models, with models implemented in PyTorch; KFM was thus programmed in the same environment. Note that PyTorch FM developers have selected hyper-parameters and fine-tuned models based on empirical testing so that the values recommended in PyTorch FM may differ from those advocated by a model's developers (all models were trained for 20 epochs with the ADAM optimiser). The intention is therefore not to make absolute statements in terms of respective performances, since recommender systems are generally built with specific applications in mind, but rather to show that KFMs perform well across the board and are versatile to accommodate different data regimes. The average-under-the-curve ("AUC"), log loss[6], average time per epoch and the number of parameters for each model, including embeddings, were computed over each dataset, cf. Table 3.2.

Key Observations A few remarks are in order. First, KFM performs well across all tasks (cf. Table 3.3), which shows that it can tackle very sparse datasets such as MovieLens, as well as denser datasets

[6] We choose the log loss, also known as the cross-entropy loss, as it is the gold standard in machine learning and is used to report most of the results in the literature. However, as per the previous chapter, one may wish to tune the loss function itself.

Table 3.3: Aggregated summary over the four datasets, across various models.

Model	Average AUC	Median AUC	Average loss	Median loss	Average rank	Median rank
LR	78.33%	79.06%	0.482	0.494	17.8	18.5
FM	80.29%	80.19%	0.461	0.471	10.0	10.0
FFM	80.27%	80.27%	0.462	0.470	9.3	10.0
HOFM	80.69%	80.71%	0.458	0.465	8.0	8.0
AFI	79.51%	79.47%	0.547	0.476	9.5	8.0
AFM	79.53%	79.59%	0.470	0.477	15.5	16.0
AFN	79.64%	79.21%	0.485	0.469	11.3	10.5
DCN	79.95%	80.06%	0.470	0.479	11.3	10.5
DFM	80.17%	79.57%	0.464	0.470	10.3	9.0
FNFM	80.19%	80.27%	0.466	0.478	8.5	7.5
FNN	79.85%	79.96%	0.470	0.480	12.8	13.5
NCF	NA	NA	NA	NA	NA	NA
NFM	78.97%	79.00%	0.530	0.478	15.0	14.5
IPNN	79.72%	79.40%	0.528	0.473	9.5	8.0
OPNN	79.89%	79.57%	0.505	0.471	6.5	5.0
WD	79.62%	79.79%	0.472	0.480	15.3	15.0
XDFM	79.54%	79.38%	0.487	0.475	10.3	11.0
KFM	80.85%	80.74%	0.450	0.446	3.5	3.0

such as Avazu and Criteo. Second, as can be checked in Table 3.2, KFM has a small number of parameters to fit, generally very close to FM, while outperforming it as the size and complexity of data grow. KFM is thus a versatile framework with a good trade-off in terms of performance and computational complexity (and training time, see for instance Figure 3.4 which depicts the performance of models in terms of area-under-the-curve (AUC) and number of parameters). Indeed, Figure 3.4 shows that state-of-the-art statistical performance can be achieved with a relatively small number of parameters, but that this result is highly dependent on the chosen architecture (i.e., not all models are able to achieve a similar accuracy with a given computational budget).

3.7 Conclusion

In this chapter, Kernel Factorisation Machines were introduced. KFMs are flexible models allowing the use of well-known kernel techniques (and are thus capable of approximating complex functions in a given reproducing kernel Hilbert space) while mitigating the drawbacks of these methods, such as the curse of dimensionality. Embeddings, inducing points and kernel design have been shown to be powerful avenues to design recommender systems on different types of data (from sparse to dense). It is worth mentioning that KFMs can recover a number of existing models, such as Factorisation Machines (Rendle, 2010) or Higher-Order Factorisation Machines (Blondel et al., 2016), which themselves encompass techniques such as matrix factorisation or (generalised) singular value decomposition (“SVD++”). Further research points to fine-tuning embeddings and kernels (for example by using deep neural kernels).

Part II

Fair Machine Learning

Chapter 4

On Learning with Fairness Trade-Offs

This chapter is the result of an initial working paper written by F.B.G., which was further expanded into a longer article co-authored with Islam Utyagulov. F.B.G. conceived of the presented ideas, developed the theoretical aspects, designed the experiments, built the initial code, and wrote the manuscript. I.U. contributed to the code base and ran the numerical experiments. Both authors discussed the results and commented on the manuscript.

Research Objectives

Since machine learning systems were able to reach society at large in many different aspects, ethical debates have arisen regarding the adoption of such technologies as well as the impact of their results. Machine Learning *fairness* is thus a recently established discipline whose main goal is to ensure that algorithms do not stem from biased data or produce outcomes that treat users unfavourably on the basis of attributes such as gender or race. As a result, a plethora of "debiasing" algorithms have been proposed to remove such biases. However, empirical investigations have revealed that the former could have poor out-of-sample performance. This chapter thus investigates the properties of a broad array of (partially) debiasing algorithms and demonstrates –on simulated and real-life data– some of their limitations. It also proposes some practical criteria to assess their suitability *ex ante*.

4.1 Introduction

Fairness in Machine Learning Machine learning ("ML") has become an increasingly important tool for decision-making in many domains, from finance and healthcare to criminal justice and hiring. However, there is growing concern about the potential for ML systems to perpetuate or even amplify existing biases and discrimination, leading to unfair or discriminatory outcomes for certain groups or individuals. This has led to a growing interest in fairness in machine learning and the development of methods and tools for ensuring that ML systems are fair and equitable.

Fairness in ML is important for a number of reasons. First, decisions made by ML systems can have significant impacts on people's lives, from access to credit and healthcare to employment and housing opportunities. These decisions should not be biased against certain groups or individuals and should be made in a fair and equitable way. Second, fairness is a core principle of many legal and ethical frameworks, including anti-discrimination laws and human rights conventions. ML systems that are biased or discriminatory may violate these principles and may be subject to legal or ethical challenges.

However, achieving fairness in ML is difficult for several reasons. ML algorithms are trained on historical data, which may reflect or even amplify biases and discrimination that exist in society. Even if the algorithms are designed to be fair, they may still reflect or perpetuate underlying biases in the data. Moreover, fairness in ML is a complex and multifaceted concept, with different notions of fairness that may conflict. For example, ensuring statistical parity (i.e., equal representation of different groups in the outcomes) may come at the cost of individual fairness (i.e., treating similar individuals similarly). Balancing these different notions of fairness requires careful consideration of the trade-offs among them and the specific context and goals of the ML system. Overall, achieving fairness in ML is a challenging but important goal and requires a multidisciplinary approach that draws on insights from computer science, statistics, law, ethics, and social science.

Debiasing ML models in practice Reducing bias amongst groups in an algorithm requires a number of steps; first, specify the fairness definitions (and thus fairness metrics) that apply; second, encode them with a penalty to measure the discrepancy between outcomes and the perfect fairness scenario; third, choose the trade-off between the original statistical goal and fairness constraints, and, fourth, pick a method to debias (at least partially) the model.

There are several obstacles to this programme. Indeed, debiasing may come at a cost (Rodolfa et al., 2020), but there are also theoretical reasons behind this claim. Jointly tackling multiple fairness definitions is usually difficult if not impossible without a decrease in model performance due to a series of impossibility theorems (Chouldechova, 2017a; Kim et al., 2020a; Kleinberg et al., 2017; Pleiss et al., 2017). Note that these results apply not only to fairness-performance but also to fairness-fairness trade-offs. A brief literature review (e.g., (Biswas & Rajan, 2020; Friedler et al., 2019)) reveals that debiasing methods often overfit on the training set and that their outcomes can vary according to the train/test set split. Recently (Agrawal et al., 2020) have shown empirically that classifiers can exhibit a mixture of improved or worsened fairness, along with improved or worsened statistical performance, after using widely available bias mitigating techniques. The authors hinted at the fact that out-of-sample generalisation error could be responsible for the lack of guaranteed improvement in either fairness or accuracy.

Research has recently been undertaken to tackle learnability and generalisation in fairness-related problems. In particular, (Oneto et al., 2020; Oneto et al., 2020) derive provable probabilistic upper bounds in a fairness setting where one estimates an overall statistical loss and monitors disparities across categories. In (Chen et al., 2018), the authors have applied bias-variance decomposition techniques to disentangle the sources of unfairness in a classifier. While this is not directly related to this present work, the risk decomposability principle is fruitful in both setups. Finally, in (Agrawal et al., 2020), a limiting distribution is established in the simple case where there is a partial requirement involving a fairness metric with two categories.

In this chapter, the focus lies on the situation of supervised learning where category membership is known, and no uncertainty is present around the specification of statistical losses and fairness metrics. Fairness trade-offs (as in (Kim et al., 2020a) for instance) are considered in a partial debiasing setting, accounting for performance–fairness and fairness–fairness trade-offs. In particular, given the recent empirical results in (Agrawal et al., 2020; Donini et al., 2018), the present objective is theoretical in nature, as the aim is to explain –to some extent– why learning fairness trade-offs often does not generalise well.

Contributions. The novelty of this work consists of the application of known techniques such as Probably Approximately Correct ("P.A.C.") inequalities and the central limit theorem to the problem of learning under fairness constraints:

- First, a Probably Approximately Correct framework for fairness trade-offs is developed, generalising results from (Donini et al., 2018; Oneto et al., 2020) and tackling explicitly *partial debiasing* (see (Agrawal et al., 2020; Kim et al., 2020a)).

- Second, it is shown that certain properties of the probabilistic upper bounds lead to the need for sample-efficient bias mitigation techniques. In particular, a new quantity, Z_S, which can be understood as a measure of sample concentration, plays a crucial role.

- Third, building on (Agrawal et al., 2020) an asymptotic framework with a known limiting distribution is developed; this is useful as P.A.C. bounds may not always be very sharp.

- Fourth, in both frameworks, decomposability of the learning risk is introduced, expressed as an upper bound in the P.A.C. realm and a limiting variance in the asymptotic one.

While there is no in-depth empirical application, as this has been done in other papers, a simple and realistic example is provided, showing the pitfalls of small sample size or class imbalance.

4.2 Fairness metrics and loss functions

4.2.1 Set-up and definitions

In this chapter, $s = 1, \cdots, C$ is a categorical (protected) attribute, $\mathbf{x} \in \mathscr{X} \subset \mathbb{R}^d$ is a set of non-protected features excluding s. $y \in \{-1, +1\}$ is a binary outcome variable and $\hat{y} \in \{-1, +1\}$ is an estimator for y. Finally, $z = (\mathbf{x}, y, s)$. Note that $\hat{y}$ is derived from a learner $h \in \mathscr{H}$, where $\mathscr{H}$ is a given functional space. Furthermore, define S to be the in-sample (training) empirical distribution, and D to refer to the true distribution.

Throughout this chapter, the case of an overall objective as a statistical performance objective, L^0, is considered, plus a fairness loss functional ϕ. The trade-off is tuned by a hyper-parameter $\lambda \geq 0$.

As usual, the aim is to minimise the overall objective, i.e., minimise the statistical loss and the lack of fairness. This setup is typical for partial debiasing and can be found in (Kim et al., 2020a).

Definition 4. The *overall objective* or *fairness trade-off* for a given $h \in \mathscr{H}$ is defined as

$$\begin{aligned}\mathscr{L}_D(h) &= L^0_D(h) + \lambda\phi\left(L^+_{D,1}(h), L^-_{D,1}(h), \cdots, L^+_{D,C}(h), L^-_{D,C}(h)\right)\\ &= L^0_D(h) + \lambda\phi(\mathbf{L}^{\pm}_D(h)),\end{aligned}$$

where we have used the standard notations (see (Shalev-Shwartz & Ben-David, 2014)):

$$\begin{aligned}L^0_D(h) &= \mathbb{E}\left[\ell^0(h,z)\right]\\ L^+_{D,a}(h) &= \mathbb{E}\left[\ell^+(h,z)|s=a, y=1\right]\\ L^-_{D,a}(h) &= \mathbb{E}\left[\ell^-(h,z)|s=a, y=-1\right]\end{aligned}$$

and the vector notation $\mathbf{L}^{\pm}_D(h) = \left[L^+_{D,1}(h), L^-_{D,1}(h), \cdots, L^+_{D,C}(h), L^-_{D,C}(h)\right]^T$, for some functions ℓ^0, ℓ^+ and ℓ^-.

Remark 9. In the whole chapter, it is assumed that there exists a uniform bound $B > 0$ on all functions ℓ, i.e., $|\ell(h,z)| \leq B$ for all h and z. ℓ refers to any such function in what follows.

Two particular cases are of interest. If $\lambda = 0$, then the overall objective boils down to the usual risk minimisation problem. If $\ell^0 = 0$ or $\lambda \to +\infty$, then the trade-off becomes a fairness constraint.

Finally, define the empirical counterpart by considering the empirical distribution rather than the true distribution. With the additional notations $\mathscr{N}^{\pm}_a = \{i \in \{1, \cdots, n\}; s_i = a, y_i = \pm 1\}$, $|\mathscr{N}^{\pm}_a| = n^{\pm}_a$, the corresponding *sample* versions can be defined as $L^0_S(h) = \frac{1}{n}\sum_{i=1}^n \ell^0(h,z_i)$, $L^+_{S,a}(h) = \frac{1}{n^+_a}\sum_{i\in\mathscr{N}^+_a} \ell^+(h,z_i)$ and $L^-_{S,a}(h) = \frac{1}{n^-_a}\sum_{i\in\mathscr{N}^-_a} \ell^-(h,z_i)$, where $n_a = n^+_a + n^-_a$, $n = \sum_{a=1}^C n_a$, leading to

$$\mathscr{L}_S(h) = L^0_S(h) + \lambda\phi\left(\mathbf{L}^{\pm}_S(h)\right) \tag{4.1}$$

For the sake of simplicity, we will generally refer to $\phi(\mathbf{L}^{\pm}_T(h))$ as $\phi_T(h)$ for $T = D, S$.

4.2.2 Fairness definitions and metrics

One can find multiple technical definitions of fairness in the literature; they have been reviewed in various papers (Agrawal et al., 2020; Berk et al., 2018; Kim et al., 2020a; Narayanan, 2018; Verma & Rubin, 2018). An overview of the most frequent metrics is offered in Table 4.1.

Note that all fairness metrics considered here can be expressed as an equality requirement on probabilities, hence on the expectation of indicator variables. These fall within the present framework. However, this framework can also handle other types of functions.

Table 4.1: Frequent fairness definitions and their probabilistic formulation.

Fairness metric	Reference	Equality requirement
Equalised false omission rate	(Berk et al., 2018)	$\mathbb{P}(y=1\|\hat{y}=-1,s=a)=\mathbb{P}(y=1\|\hat{y}=-1)$
Predictive parity	(Chouldechova, 2017a)	$\mathbb{P}(y=1\|\hat{y}=1,s=a)=\mathbb{P}(y=1\|\hat{y}=1)$
Demographic parity	(Calders & Verwer, 2010)	$\mathbb{P}(\hat{y}=1\|s=a)=\mathbb{P}(\hat{y}=1)$
Equalised false negative rate	(Chouldechova, 2017a)	$\mathbb{P}(\hat{y}=-1\|y=1,s=a)=\mathbb{P}(\hat{y}=-1\|y=1)$
Predictive equality	(Chouldechova, 2017a)	$\mathbb{P}(\hat{y}=1\|y=-1,s=a)=\mathbb{P}(\hat{y}=1\|y=-1)$
Equality of opportunity	(Hardt et al., 2016)	$\mathbb{P}(\hat{y}=1\|y=1,s=a)=\mathbb{P}(\hat{y}=1\|y=1)$
Equalised odds	(Hardt et al., 2016)	$\mathbb{P}(\hat{y}=1\|y=y',s=a)=\mathbb{P}(\hat{y}=1\|y=y')$

4.2.3 Fairness loss functionals

In addition to picking one or multiple fairness definitions, ϕ is specified to measure the discrepancy from perfect fairness, the ideal case.

A first approach, similar to the Calders-Verwer gap (Calders & Verwer, 2010), consists of looking at the discrepancy between two categories a and a':

$$\Delta^{L,\pm}_{T,a,a'}(h) = L^{\pm}_{T,a}(h) - L^{\pm}_{T,a'}(h),$$

for $T \in \{D,S\}$. It is worth 0 under perfect fairness but is asymmetric and thus requires defining an advantaged (or benchmark) category a.

To avoid the issue of asymmetry, one can follow (Oneto et al., 2020), and define ϕ as the sum of all absolute discrepancies across categories:

$$\phi(\mathbf{L}^{\pm}_{T}(h)) = \sum_{a\neq a'} \left\{ \left| L^{+}_{T,a} - L^{+}_{T,a'} \right| + \left| L^{-}_{T,a} - L^{-}_{T,a'} \right| \right\}, \tag{4.2}$$

for $T = D,S$. Notice that, thanks to the reverse triangle inequality, the function ϕ is Lipschitz continuous. One could naturally weigh the various contributions to this fairness function and use a norm other than L^1. Other functions involving ratios or relative differences, are also possible.

Finally, in (Kim et al., 2020a), the authors consider a convex combination of multiple loss functions $\phi^{(1)},\cdots,\phi^{(M)}$:

$$\phi(\mathbf{L}^{\pm}_{T}(h)) = \sum_{j=1}^{M} \lambda_j \phi^{(j)}\left(\mathbf{L}^{\pm}_{T}(h)\right).$$

Note that, if the $\phi^{(j)}$'s are Lipschitz-continuous (with respect to the loss vector $\mathbf{L}^{\pm}_{T}(h)$) with respective Lipschitz constant $K^{(j)}$, then the overall loss ϕ is itself Lipschitz-continuous with constant $K^{\phi} = \sum_{j=1}^{M} \lambda_j K^{(j)}$.

4.3 Learning fairness trade-offs

In this section, the learning problem is considered from different angles. A bound can be derived by considering the statistical loss L^0, based on the Rademacher complexity observed in each category. To do so, the statistical learning toolbox provides useful tools, such as Rademacher complexities.

Learning the fairness objective can then be considered per se, i.e., the generalisation properties of $\phi_S(h)$ studied. Finally, the two are brought together, and probabilistic upper bounds on fairness trade-off generalisation are derived.

4.3.1 Learning the statistical loss

First, focus on the *statistical* component of the loss, i.e., L_T^0 for $T \in \{S, D\}$. Much of this chapter is based on theory of bounds derived from Rademacher complexities, introduced in (Bartlett & Mendelson, 2002) and surveyed in (Boucheron et al., 2005). Recent textbooks such as (Mohri et al., 2012; Shalev-Shwartz & Ben-David, 2014) provide excellent introductions to the topic. Recall the definition of Rademacher complexities, as indicated in (Boucheron et al., 2005):

Definition 5. The Rademacher complexity (or average) of a function ℓ is given by

$$R(\ell \circ \mathscr{H} \circ \mathscr{S}) = \mathbf{E}\left[\sup_{h\in\mathscr{H}} \frac{1}{n}\left|\sum_{i=1}^{n} \sigma_i \ell(h, z_i)\right|\right], \tag{4.3}$$

where the σ_i's are independent and identically distributed Rademacher random variables (i.e., $\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma_i = -1] = 1/2$), $\mathscr{S}$ is the sample of points $(z_1, \cdots, z_n)$ and $\mathscr{H}$ is the space of possible candidate functions h.

Note that definitions can slightly vary (depending on the presence or absence of the absolute value in the definition). Still, all downstream results are qualitatively similar, up to some multiplicative constants. While the Rademacher complexity is defined on a given sample $\mathscr{S}$, we make the assumption in what follows that the observations z_i are drawn independently from the data distribution D.

To make this more concrete, a simple (but usual) case (as described in (Shalev-Shwartz & Ben-David, 2014)) can be considered. Suppose that almost surely $\|\mathbf{x}\|_2 \leq R$. In addition, let $\mathscr{H} = \{\mathbf{w}; \|\mathbf{w}\|_2 \leq R'\}$ and assume that the loss function ℓ is of the type

$$\ell(\mathbf{w}, (\mathbf{x}, y)) = \rho\left(\mathbf{w}^T\mathbf{x}, y\right),$$

where $|\rho|$ is bounded by B and is L^ρ-Lipschitz. Then, almost surely,

$$R(\ell \circ \mathscr{H} \circ \mathscr{S}) \leq \frac{L^\rho R R'}{\sqrt{n}}.$$

Rademacher complexities are key to establishing learnability bounds and lead to fundamental results in statistical learning theory. In particular, the following standard result, e.g., see Theorem 3.2 in (Boucheron et al., 2005), is repeatedly used.

Proposition 11. *With probability at least* $1 - \delta$*, it holds*

$$\sup_{h\in\mathscr{H}} |L_D(h) - L_S(h)| \leq 2R(\ell \circ \mathscr{H} \circ \mathscr{S}) + B\sqrt{\frac{2\log\frac{2}{\delta}}{n}}. \tag{4.4}$$

It will also be useful to introduce *conditional* Rademacher complexities that will be used throughout this chapter. In particular, since a partition of the sample in terms of categories is available

$$\mathscr{S} = \bigcup_{a=1}^{C} \mathscr{N}_a, \tag{4.5}$$

Rademacher complexity of that particular sample can be considered:

$$R(\ell \circ \mathscr{H} \circ \mathscr{N}_a) = \mathbf{E}\left[\sup_{h\in\mathscr{H}} \frac{1}{n_a}\left|\sum_{i\in\mathscr{N}_a} \sigma_i \ell(h, z_i)\right|\right]. \tag{4.6}$$

Lemma 1. *The sample Rademacher complexity can be bounded from above by the weighted sum of conditional Rademacher complexities:*

$$R(\ell \circ \mathscr{H} \circ \mathscr{S}) \leq \sum_{a=1}^{C} \frac{n_a}{n} R(\ell \circ \mathscr{H} \circ \mathscr{N}_a), \tag{4.7}$$

where $R(\ell \circ \mathscr{H} \circ \mathscr{N}_a) = \mathbf{E}\left[\sup_{h\in\mathscr{H}} \frac{1}{n_a}\left|\sum_{i\in\mathscr{N}_a} \sigma_i \ell(h, z_i)\right|\right]$.

Proof. This comes directly from the sub-additivity of the absolute value and the supremum. □

One can further interpret the non-negative gap $\sum_{a=1}^{C} \frac{n_a}{n} R(\ell \circ \mathscr{H} \circ \mathscr{N}_a) - R(\ell \circ \mathscr{H} \circ \mathscr{S})$ as a diversification benefit amongst categories. Finally, this leads to a proposition leveraging the results obtained so far:

Proposition 12. *With probability at least* $1-\delta$,

$$\sup_{h\in\mathscr{H}} \left|L_D^0(h) - L_S^0(h)\right| \leq 2\sum_{a=1}^{C} \frac{n_a}{n} R(\ell^0 \circ \mathscr{H} \circ \mathscr{N}_a) + B\sqrt{\frac{2\log\frac{2}{\delta}}{n}}. \tag{4.8}$$

4.3.2 Learning fairness requirements

It is now possible to proceed to the *fairness* learning part; as previously, a bound on the *distribution* fairness loss given the *sample* fairness loss, i.e., a probabilistic bound on the difference $\phi_D(h) - \phi_S(h)$, is sought after.

Proposition 13. *Under the assumption that* ϕ *is* K^ϕ*-Lipschitz, it holds, with probability at least* $1-\delta$, *that*

$$\begin{aligned} \sup_{h\in\mathscr{H}} |\phi_D(h) - \phi_S(h)| \quad &\leq \quad 2K^\phi \sum_{a=1}^{C} R(\ell^+ \circ \mathscr{H} \circ \mathscr{N}_a^+) + R(\ell^- \circ \mathscr{H} \circ \mathscr{N}_a^-) \\ &\quad + K^\phi B Z_S \sqrt{\frac{2\log\frac{4C}{\delta}}{n}}. \end{aligned}$$

with $Z_S := \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^+}} + \sqrt{\frac{n}{n_a^-}}$.

Proof. Using the hypothesis that ϕ is K^ϕ-Lipschitz, it comes

$$|\phi_D(h) - \phi_S(h)| \leq K^\phi \sum_{a=1}^{C} \left| L^+_{D,a}(h) - L^+_{S,a}(h) \right| + \left| L^-_{D,a}(h) - L^-_{S,a}(h) \right|.$$

For any $\delta' \in (0,1)$ and any $a = 1, \cdots, C$ it holds

$$\sup_{h \in \mathscr{H}} \left| L^\pm_{D,a}(h) - L^\pm_{S,a}(h) \right| \leq 2R(\ell^\pm \circ \mathscr{H} \circ \mathscr{N}_a^{\pm}) + B\sqrt{\frac{2\log\frac{2}{\delta'}}{n_a^\pm}},$$

whereby

$$\begin{aligned} \sup_{h \in \mathscr{H}} |\phi_D(h) - \phi_S(h)| \quad &\leq \quad 2K^\phi \sum_{a=1}^{C} R(\ell^+ \circ \mathscr{H} \circ \mathscr{N}_a^{+}) + R(\ell^- \circ \mathscr{H} \circ \mathscr{N}_a^{-}) \\ &\quad + K^\phi B \sum_{a=1}^{C} \sqrt{\frac{2\log\frac{2}{\delta'}}{n_a^+}} + \sqrt{\frac{2\log\frac{2}{\delta'}}{n_a^-}}, \end{aligned}$$

with probability at least $1 - 2C\delta'$, thanks to the union bound. To have $1 - \delta = 1 - 2C\delta'$, one can simply pick $\delta' = \delta/(2C)$. Now, the last term can be rewritten as

$$\begin{aligned} \sum_{a=1}^{C} \sqrt{\frac{2\log\frac{2}{\delta'}}{n_a^+}} + \sqrt{\frac{2\log\frac{2}{\delta'}}{n_a^-}} \quad &= \quad \sqrt{\frac{2\log\frac{2}{\delta'}}{n}} \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^+}} + \sqrt{\frac{n}{n_a^-}} \\ &= \quad Z_S \sqrt{\frac{2\log\frac{2}{\delta'}}{n}}, \end{aligned}$$

hence the final result. □

Remark 10. The upper bound is interesting in itself from a qualitative standpoint. $Z_S = \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^+}} + \sqrt{\frac{n}{n_a^-}}$ is the sum of the square root of empirical class probabilities n_a^+. Thus, having very small empirical probabilities can lead to very high bounds, which implies that enforcing fairness in the presence of underrepresented groups does not necessarily generalise easily. In some sense, Z_S is a measure of concentration of the sample $\mathscr{S}$ and is minimised when $n_a^+ = n_a^- = n_{a'}^+ = n_{a'}^-$ for all a, a'.

4.3.3 Simultaneous learning of statistical performance and fairness

Probabilistic bounds have been established on both the statistical performance criterion and the fairness penalty. However, both should be studied jointly and the behaviour of the chosen fairness trade-off determined. This trade-off is very flexible and accommodates multiple situations.

4.3.3.1 Bounding loss and fairness

First, it is possible to determine a probabilistic upper bound on $L_D^0(h) - L_S^0(h)$ and $\phi_D(h) - \phi_S(h)$ jointly. By a simple application of the union bound, the following result is obtained:

Proposition 14. *With probability at least* $1-\delta$*, it holds* jointly *that*

$$\begin{aligned}
\sup_{h\in\mathscr{H}} \left|L_D^0(h)-L_S^0(h)\right| &\leq 2R(\ell^0\circ\mathscr{H}\circ\mathscr{S})+B\sqrt{\frac{2\log\frac{4}{\delta}}{n}}\\
\sup_{h\in\mathscr{H}} \left|\phi_D(h)-\phi_S(h)\right| &\leq 2K^\phi\sum_{a=1}^{C} R(\ell^+\circ\mathscr{H}\circ\mathscr{N}_a^+)+R(\ell^-\circ\mathscr{H}\circ\mathscr{N}_a^-)\\
&\quad +K^\phi BZ_S\sqrt{\frac{2\log\frac{8C}{\delta}}{n}}.
\end{aligned}$$

4.3.3.2 Bounding the trade-off

As mentioned above, various impossibility results and the empirical findings showing that statistical performance tends to decrease as fairness requirements increase have highlighted the interest in partial debiasing methods. It is, however, essential to determine the generalisation properties that such objectives can lead to. A similar analysis on the trade-off objective $\mathscr{L}_D(h)$ can be performed using the same arguments to get an upper bound.

Proposition 15. *With probability at least* $1-\delta$*, it holds*

$$\begin{aligned}
\sup_{h\in\mathscr{H}} \left|\mathscr{L}_D(h)-\mathscr{L}_S(h)\right| &\leq 2R(\ell^0\circ\mathscr{H}\circ\mathscr{S})+B\sqrt{\frac{2\log\frac{4}{\delta}}{n}}\\
&\quad +2\lambda K^\phi\sum_{a=1}^{C} R(\ell^+\circ\mathscr{H}\circ\mathscr{N}_a^+)+R(\ell^-\circ\mathscr{H}\circ\mathscr{N}_a^-)\\
&\quad +\lambda K^\phi BZ_S\sqrt{\frac{2\log\frac{8C}{\delta}}{n}}\\
&\leq 2\sum_{a=1}^{C}\left\{\frac{n_a^+}{n}R(\ell^0\circ\mathscr{H}\circ\mathscr{N}_a^+)+\frac{n_a^-}{n}R(\ell^0\circ\mathscr{H}\circ\mathscr{N}_a^-)\right.\\
&\quad \left.+\lambda K^\phi R(\ell^+\circ\mathscr{H}\circ\mathscr{N}_a^+)+\lambda K^\phi R(\ell^-\circ\mathscr{H}\circ\mathscr{N}_a^-)\right\}\\
&\quad +B\left(1+\lambda K^\phi Z_S\right)\sqrt{\frac{2\log(8C/\delta)}{n}}.
\end{aligned}$$

The upper bound can be understood as the sum of individual contributions plus the usual $O\left(\sqrt{\log(1/\delta)/n}\right)$ factor.

A related inequality can also be established, linking the distribution value of the trade-off under the distribution optimum and the sample optimum. Let us denote by $h_T^* = \arg\inf_{h\in\mathscr{H}}\mathscr{L}_T(h)$ for $T\in\{D,S\}$ an optimal classifier derived on either the underlying distribution or the sample distribution of the objectives $\mathscr{L}_T$. This is a crucial issue for learnability of fairness as it provides some guarantees on how far the optimal trade-off is from the one obtained by picking the optimum computed on the sample $\mathscr{S}$. This result provides a fair pendant to the usual case in statistical learning (see, for instance, Theorem 26.5 in (Shalev-Shwartz & Ben-David, 2014)).

Theorem 4. *With probability at least $1-\delta$, it holds*

$$0 \leq \mathscr{L}_D(h_S^*) - \mathscr{L}_D(h_D^*) \leq UB_{\mathscr{S}} + B\left(1+\lambda Z_S K^\phi\right)\frac{3}{\sqrt{2}}\sqrt{\frac{\log(16C/\delta)}{n}}, \tag{4.9}$$

where $h_T^* = \arg\inf_{h\in\mathscr{H}} \mathscr{L}_T(h)$ for $T \in \{D,S\}$, and $UB_{\mathscr{S}} := 2\sum_{a=1}^C \left\{\frac{n_a^+}{n}R(\ell^0 \circ \mathscr{H} \circ \mathscr{N}_a^+) + \frac{n_a^-}{n}R(\ell^0 \circ \mathscr{H} \circ \mathscr{N}_a^-) + \lambda K^\phi R(\ell^+ \circ \mathscr{H} \circ \mathscr{N}_a^+) + \lambda K^\phi R(\ell^- \circ \mathscr{H} \circ \mathscr{N}_a^-)\right\}$.

Proof. For the sake of clarity, the proof is divided into different steps.

Step 1. Start by rewriting

$$\begin{aligned}\mathscr{L}_D(h_S^*) - \mathscr{L}_D(h_D^*) &= \mathscr{L}_D(h_S^*) - \mathscr{L}_S(h_S^*) + \mathscr{L}_S(h_S^*) - \mathscr{L}_S(h_D^*) + \mathscr{L}_S(h_D^*) - \mathscr{L}_D(h_D^*) \\ &\leq \mathscr{L}_D(h_S^*) - \mathscr{L}_S(h_S^*) + \mathscr{L}_S(h_D^*) - \mathscr{L}_D(h_D^*),\end{aligned}$$

since, by definition, $\mathscr{L}_S(h_S^*) - \mathscr{L}_S(h_D^*) \leq 0$.

Step 2. Now, $\mathscr{L}_D(h_S^*) - \mathscr{L}_S(h_S^*) \leq \sup_{h\in\mathscr{H}} |\mathscr{L}_D(h) - \mathscr{L}_S(h)|$ on the one hand, and, on the other hand, it holds

$$\begin{aligned}|\mathscr{L}_S(h_D^*) - \mathscr{L}_D(h_D^*)| \leq\ & \left|L_S^0(h_D^*) - L_D^0(h_D^*)\right| \\ & + \lambda K^\phi \sum_{a=1}^C \left\{\left|L_{D,a}^+(h_D^*) - L_{S,a}^+(h_D^*)\right| + \left|L_{D,a}^-(h_D^*) - L_{S,a}^-(h_D^*)\right|\right\}.\end{aligned}$$

Step 3. The key insight is to notice that the right-hand side only involves h_D^*, which does *not* depend on the sample $\mathscr{S}$; hence one can apply the Hoeffding inequality directly. Thus, by Hoeffding's inequality, for each a, with probability at least $1-\delta''$,

$$\left|L_{D,a}^\pm(h_D^*) - L_{S,a}^\pm(h_D^*)\right| \leq B\sqrt{\frac{\log(2/\delta'')}{2n_a^\pm}}.$$

Similarly, with probability at least $1-\delta''$,

$$\left|L_S^0(h_D^*) - L_D^0(h_D^*)\right| \leq B\sqrt{\frac{\log(2/\delta'')}{2n}}.$$

Thus, with probability at least $1-(2C+1)\delta''$, one infers that

$$\begin{aligned}|\mathscr{L}_S(h_D^*) - \mathscr{L}_D(h_D^*)| &\leq B\sqrt{\frac{\log(2/\delta')}{2n}} \\ &\quad + \lambda B K^\phi \sum_{a=1}^C \sqrt{\frac{\log(2/\delta'')}{2n_a^+}} + \sqrt{\frac{\log(2/\delta'')}{2n_a^-}} \\ &\leq B\left(1+\lambda Z_S K^\phi\right)\sqrt{\frac{\log(2/\delta'')}{2n}}.\end{aligned}$$

Step 4. From the previous proposition, it follows that with probability at least $1-\delta'$

$$\sup_{h\in\mathscr{H}} |\mathscr{L}_D(h)-\mathscr{L}_S(h)| \leq \mathrm{UB}_{\mathscr{S}} + B\left(1+\lambda Z_S K^{\phi}\right)\sqrt{\frac{2\log(8C/\delta')}{n}}.$$

Finally, by choosing $\delta' = \delta/2$ and $\delta'' = \delta/(2(2C+1))$,

$$\begin{aligned}\mathscr{L}_D(h_S^*)-\mathscr{L}_D(h_D^*) &\leq \mathrm{UB}_{\mathscr{S}} + B\left(1+\lambda Z_S K^{\phi}\right)\left[\sqrt{\frac{2\log(16C/\delta)}{n}}+\sqrt{\frac{\log(4(2C+1)/\delta)}{2n}}\right]\\ &\leq \mathrm{UB}_{\mathscr{S}} + B\left(1+\lambda Z_S K^{\phi}\right)\frac{3}{\sqrt{2}}\sqrt{\frac{\log(16C/\delta)}{n}},\end{aligned}$$

with probability at least $1-\delta$. □

What is particularly interesting about this result is the fact that the upper bound remains the same as in Proposition 15, and only the $O\left(\sqrt{\log(1/\delta)/n}\right)$ factor changes. It should be pointed out that this result is *practical* in nature in the sense that one would use h_S^* in practice but would still look for out-of-sample generalisation, hence the importance of the term $\mathscr{L}_D(h_S^*)$.

4.4 Examples

This Section applies results to two particular loss functions, namely the Calders-Verwer gap and the Bayes gap.

4.4.1 Example 1: Learning disparity

Of particular interest when measuring fairness is the discrepancy that can be observed amongst different groups, for example, in terms of a loss L that captures some properties we wish to ascertain. This could be the difference in false negative rates between categories a and a', where a is usually chosen to be a reference group. This concept holds for both the true distribution and the sample distribution.

Definition 6. The (demographic) disparity, or Calders-Verwer gap (Calders & Verwer, 2010; Chen et al., 2019a), between groups a and a' with respect to loss L can be defined as

$$\Delta^L_{T,a,a'}(h) = L_{T,a}(h) - L_{T,a'}(h), \tag{4.10}$$

for $T \in \{D,S\}$.

Note that it is also quite usual to consider the absolute value of disparity, $\left|\Delta^L_{T,a,a'}(h)\right|$ indicating whether there is equality or not (and magnitude thereof), rather than giving a sign (which relies more on how the baseline category has been chosen). Indeed, the disparity is trivially asymmetric, whereas the absolute disparity is symmetric.

It is, therefore, important to establish the learnability of this widely used metric, which we do in the following proposition.

Proposition 16. *With probability at least* $1-\delta$*, the following inequality holds*

$$\begin{aligned}\sup_{h\in\mathscr{H}}\left|\Delta^L_{D,a,a'}(h)-\Delta^L_{S,a,a'}(h)\right| &\leq 2\left\{R(\ell\circ\mathscr{H}\circ\mathscr{N}_a)+R(\ell\circ\mathscr{H}\circ\mathscr{N}_{a'})\right\}\\ &\quad +B\sqrt{\frac{2\log\left(\frac{4}{\delta}\right)}{n_a+n_{a'}}}\left[\sqrt{\frac{n_a+n_{a'}}{n_a}}+\sqrt{\frac{n_a+n_{a'}}{n_{a'}}}\right].\end{aligned}$$

Proof. Start by observing that by sub-additivity of the sup, the following holds:

$$\sup_{h\in\mathscr{H}}\left|\Delta^L_{D,a,a'}(h)-\Delta^L_{S,a,a'}(h)\right|\leq\sup_{h\in\mathscr{H}}\left|L_{D,a}(h)-L_{S,a}(h)\right|+\sup_{h\in\mathscr{H}}\left|L_{D,a'}(h)-L_{S,a'}(h)\right|.$$

The proof is concluded thanks to Proposition 11 and the union bound. □

What this upper bound suggests is that in order to learn about the disparity, one has to learn the two categories a and a'. If one category has either a large Rademacher complexity or a low count, the upper bound will increase. This result enables us to derive bounds on quantities of interest such as $\left|\Delta^L_{D,a,a'}(h^*_S)-\Delta^L_{S,a,a'}(h^*_S)\right|$ that follows directly from Proposition 16.

4.4.2 Example 2: Learning bias

In this Section, bias is understood in a very specific (and non-fairness related) way, as in (Chen et al., 2018). In short, one can consider bias –in this context– as the difference between the loss incurred by category a when the classifier is determined by minimising the loss function over the entire sample and the loss in category a when the optimal classifier is derived on a standalone basis (i.e., the classifier explicitly takes into the account the attribute a).

If it is possible to use the characteristics s directly in the model, then an optimal classifier can be calibrated on each category:

$$h^*_D(\mathbf{x},s)=\sum_{a=1}^{C}\mathbf{1}_{\{s=a\}}h^*_{D,a}(\mathbf{x}). \tag{4.11}$$

However, the characteristic s is usually unavailable or cannot be included as a feature (for instance, to avoid disparate treatment), so that –in general– $h_D(\mathbf{x},s)=h^*_D(\mathbf{x})$ for all $s=1,\cdots,C$. Consequently, the Bayes gap can be defined as follows:

Definition 7. The *Bayes gap* for category a is given by

$$\Gamma_{T,a}=L_{T,a}(h^*_S)-L_{T,a}(h^*_{S,a}), \tag{4.12}$$

for $T\in\{D,S\}$.

In particular, $\Gamma_{S,a}\geq 0$. $\Gamma_{\cdot,a}$ represents the added loss incurred by category a due to the consideration of other categories while selecting the classifier h.

Proposition 17. *The true Bayes gap,* $\Gamma_{D,a} = L_{D,a}(h_S^*) - L_{D,a}(h_{S,a}^*)$ *can be learnt as*

$$|\Gamma_{D,a} - \Gamma_{S,a}| \leq 4R(\ell \circ \mathscr{H} \circ \mathscr{N}_a) + 2B\sqrt{\frac{2\log(2/\delta)}{2n_a}}, \tag{4.13}$$

with probability at least $1-\delta$.

Proof. The proof is fairly straightforward and decomposes the Bayes gap in terms of easier building blocks:

$$\begin{aligned} L_{D,a}(h_S^*) - L_{D,a}(h_{S,a}^*) \quad &= \quad L_{D,a}(h_S^*) - L_{S,a}(h_S^*) + L_{S,a}(h_S^*) - L_{S,a}(h_{S,a}^*) \\ &\quad\quad + L_{S,a}(h_{S,a}^*) - L_{D,a}(h_{S,a}^*), \end{aligned}$$

leading to

$$\Gamma_{D,a} - \Gamma_{S,a} = L_{D,a}(h_S^*) - L_{S,a}(h_S^*) + L_{S,a}(h_{S,a}^*) - L_{D,a}(h_{S,a}^*).$$

Finally, it follows that

$$\begin{aligned} |\Gamma_{D,a} - \Gamma_{S,a}| \quad &\leq \quad 2\sup_{h \in \mathscr{H}} |L_{D,a}(h) - L_{S,a}(h)| \\ &\leq \quad 4R(\ell \circ \mathscr{H} \circ \mathscr{N}_a) + 2B\sqrt{\frac{2\log(2/\delta)}{2n_a}}, \end{aligned}$$

where the second inequality holds with probability at least $1-\delta$ by Proposition 11. □

4.5 Practical consequences of P.A.C. results for trade-offs

4.5.1 Decomposing upper bounds on generalisation

One is led to decompose the overall loss in terms of each category's contributions to the overall learning upper bound.

Definition 8. The (half-)contribution –denoted by $\overline{R}_{\mathscr{N}_a}$– of each category $a = 1, \cdots, C$, to the fairness trade-off learning upper bound is

$$\begin{aligned} \overline{R}_{\mathscr{N}_a} \quad := \quad &\frac{n_a^+}{n} R(\ell^0 \circ \mathscr{H} \circ \mathscr{N}_a^+) + \frac{n_a^-}{n} R(\ell^0 \circ \mathscr{H} \circ \mathscr{N}_a^-) \\ &+ \lambda K^\phi R(\ell^+ \circ \mathscr{H} \circ \mathscr{N}_a^+) + \lambda K^\phi R(\ell^- \circ \mathscr{H} \circ \mathscr{N}_a^-). \end{aligned}$$

Remark 11. This structure is quite interesting as it shows that the first two terms representing contributions to the statistical loss upper bound are weighted by their proportions of the overall sample. Thus, even if the Rademacher complexity of the category is high, it can be counterbalanced by the fact that it will not affect the overall average loss. On the other hand, the contributions coming from the fairness part are unweighted. In this case, a low sample size would usually lead to a higher Rademacher complexity.

Now, $\overline{R}_{\mathcal{N}_a}$ can be decomposed into finer components:

$$\overline{R}_{\mathcal{N}_a} = \overline{R}^{0,+}_{\mathcal{N}_a} + \overline{R}^{0,-}_{\mathcal{N}_a} + \overline{R}^{+}_{\mathcal{N}_a} + \overline{R}^{-}_{\mathcal{N}_a}, \tag{4.14}$$

with obvious notations. One can then determine the contributions from the union of certain categories or positives or negatives.

4.5.2 Z_S as sample concentration

First, remark that the usefulness of probabilistic upper bounds comes from the fact that the ones that have been established only rely on observable quantities coming from the sample under consideration. This is powerful as a practical check, but it also suggests a way of amending the initial minimisation exercise by including the upper bound to the sample component, i.e., optimise $\mathscr{L}_D(h) + \mathrm{UB}_{\mathscr{S}}(h)$.

Second, a quantity that is ubiquitous in the present analysis (see, for instance, Theorem 4) is $Z_S = \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^+}} + \sqrt{\frac{n}{n_a^-}}$. This measure of concentration within the sample only depends on in-sample data. The more unequal the count of categories' sample sizes, the higher Z_S. This can be formalised in the following proposition:

Proposition 18. *The constant Z_S can be expressed in terms of the empirical class sample proportions:*

$$Z_S = \sum_{a=1}^{C} \left[(\widehat{p}_a^+)^{-\frac{1}{2}} + (\widehat{p}_a^-)^{-\frac{1}{2}} \right] = \frac{2C}{\sqrt{M_{-1/2}(\widehat{\mathbf{p}})}}, \tag{4.15}$$

where $M_\alpha(\widehat{\mathbf{p}})$ is the α-generalised mean [1] *of the vector $\widehat{\mathbf{p}} = (\widehat{p}_a^\pm)_a^\pm$. In addition, under the assumption that $n_a^\pm \geq 1$ for all $a = 1, \cdots, C$, and $n > 2C - 1$, the following inequality holds:*

$$(2C)^{3/2} \leq Z_S \leq (2C-1)\sqrt{n} + \sqrt{\frac{n}{n-(2C-1)}}. \tag{4.16}$$

The proof is straightforward and left to the reader. Arguably, the bounds on Z_S offer two conclusions. First, the lower bound grows super-linearly as a function of the number of categories; in particular, if the case of *intersectionality* is considered (i.e., looking at Cartesian products of p protected attributes) and supposing that each attribute has only two categories, then $C = 2^p$, leading to a lower bound worth $2^{\frac{3}{2}(p+1)}$, which increases very quickly. Second, the upper bound on Z_S converges to a *non-zero* constant as n goes to infinity. This is obviously an extreme scenario where a category can be made up of one individual only, but this stresses that class imbalance can hinder learning in the context of fairness.

This quantity, which is entirely *sample-specific* and does not depend on any model, appears to

[1] The α-generalised mean of K numbers $x_1, \cdots, x_K$ is given by $\left(\frac{1}{K}\sum_{k=1}^{K} x_k^{\alpha}\right)^{1/\alpha}$.

be of particular importance when considering fairness trade-offs from a generalisation standpoint, as it controls the sample's complexity in terms of the number of categories and class imbalance. It could be useful in practice to measure the difficulty of applying bias-mitigating techniques on given datasets.

4.6 Asymptotic regime

One of the possible issues of using P.A.C. learning in practice is that it tends to consider worst-case scenarios (e.g., by deriving probabilistic inequalities over a whole space of possible classifiers). A very different viewpoint is now adopted to mitigate this possible drawback, generalising Proposition 1 in (Agrawal et al., 2020). The main objective at hand is to derive a limiting distribution for the rescaled fairness trade-off after having trained *one* particular classifier, which is now considered fixed. In doing so, the different contributions to the limiting variance *given* the choice of classifier can be assessed. In addition to corresponding to real-life modelling, this also allows for *closed-form* results (as opposed to upper bounds in P.A.C. learning). Both approaches are –as will be shown– very complementary.

A few words about the idealised data-generating process that hypothetically produces our observations are in order. In particular, an infinite population of independently and identically distributed data points is considered, according to a mixture distribution with probability weights $p_a^\pm$. Conditional on a and $y = \pm 1$, $\mathbf{x}$ can be drawn. All draws are independent. Some additional notation will be useful: the intra-category variances are given by $\left(\sigma_a^{0,\pm}\right)^2 = \mathbb{V}_{s=a,y=\pm 1}\left[\ell^0(h,z)\right]$, $\left(\sigma_a^{\pm}\right)^2 = \mathbb{V}_{s=a,y=\pm 1}\left[\ell^\pm(h,z)\right]$. Furthermore, $K_{D,a}^\pm = \partial_{x_a^\pm}\phi_D(h)$, i.e., $K_{D,a}^\pm$ is the entry corresponding to $a^\pm$ in the gradient $\nabla\phi_D(h)$.

4.6.1 Limiting variance

The main result can now be introduced, extending (Agrawal et al., 2020) to a multi-category and multi-dimensional setting.

Theorem 5. *Let $h \in \mathscr{H}$, under the assumption that $p_a^\pm > 0$ for all a, and that $\nabla\phi_D(h)$ exists, is continuous*[2] *and non-zero, the following convergence in distribution holds:*

$$\sqrt{n}\left(\mathscr{L}_S(h) - \mathscr{L}_D(h)\right) \xrightarrow[n\to+\infty]{d} N\left(0, \mathbb{V}_{\lim}(h)\right), \tag{4.17}$$

[2] As mentioned in the proof that follows the theorem statement, this assumption can be relaxed but is kept here for simplicity's sake.

where

$$
\begin{aligned}
\mathbb{V}_{\lim}(h) \quad = \quad & \sum_{a=1}^{C} p_a^+ \left(\sigma_a^{0,+}\right)^2 + \lambda^2 \frac{(K_a^+)^2}{p_a^+} \left(\sigma_a^+\right)^2 + 2\lambda K_a^+ cov_a^+ \left(\ell^0(h,z), \ell^+(h,z)\right) \\
& + \sum_{a=1}^{C} p_a^- \left(\sigma_a^{0,-}\right)^2 + \lambda^2 \frac{(K_a^-)^2}{p_a^-} \left(\sigma_a^-\right)^2 + 2\lambda K_a^- cov_a^- \left(\ell^0(h,z), \ell^-(h,z)\right) \\
& + \sum_{a=1}^{C} p_a^+(1-p_a^+) \left(L_{D,a}^{0,+}\right)^2 + p_a^-(1-p_a^-) \left(L_{D,a}^{0,-}\right)^2 - 2p_a^+ p_a^- L_{D,a}^{0,+} L_{D,a}^{0,-} \\
& - \sum_{a \neq a'} \left(p_a^+ p_{a'}^+ L_{D,a}^{0,+} L_{D,a'}^{0,+} + p_a^- p_{a'}^- L_{D,a}^{0,-} L_{D,a'}^{0,-} \right. \\
& \left. + p_a^+ p_{a'}^- L_{D,a}^{0,+} L_{D,a'}^{0,-} + p_a^- p_{a'}^+ L_{D,a}^{0,-} L_{D,a'}^{0,+} \right)
\end{aligned}
$$

Proof. For the sake of clarity, this simple (but slightly tedious) proof is divided into multiple steps. In this proof, the assumption that ϕ is differentiable with continuous gradient is made; this is not necessary but simplifies the exposition. The reader is referred to (DasGupta, 2008) for details regarding the delta method, the central limit theorem and different types of convergence.

Step 1. Start by rewriting the difference between both true and empirical trade-offs $\mathscr{L}_S$ and $\mathscr{L}_D$ for a given $h \in \mathscr{H}$. Thanks to the Taylor formula, there exists $\xi \in [0,1]$ such that

$$
\begin{aligned}
\phi_S(h) - \phi_D(h) \quad &= \quad \phi(\mathbf{L}_S^{\pm}(h)) - \phi(\mathbf{L}_D^{\pm}(h)) \\
&= \quad \nabla\phi(\xi \mathbf{L}_D^{\pm}(h) + (1-\xi)\mathbf{L}_S^{\pm}(h))^T \left(\mathbf{L}_S^{\pm}(h) - \mathbf{L}_D^{\pm}(h)\right)
\end{aligned}
$$

Denote by $K_{S,a}^{\pm}$ the partial differential of ϕ with respect to $x_a^{\pm}$, i.e., $K_{S,a}^{\pm} = \partial_{x_a^{\pm}} \phi(\xi \mathbf{L}_D^{\pm}(h) + (1-\xi)\mathbf{L}_S^{\pm}(h))^T \left(\mathbf{L}_S^{\pm}(h) - \mathbf{L}_D^{\pm}(h)\right)$. Some algebra then yields

$$
\begin{aligned}
& \sqrt{n}\left(\mathscr{L}_S(h) - \mathscr{L}_D(h)\right) \\
& \quad = \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^+}} \sqrt{n_a^+} \left(p_a^+ \left[L_{S,a}^{0,+}(h) - L_{D,a}^{0,+}(h) \right] + \lambda K_{S,a}^+ \left[L_{S,a}^+(h) - L_{D,a}^+(h) \right] \right) \\
& \quad + \sum_{a=1}^{C} \sqrt{\frac{n}{n_a^-}} \sqrt{n_a^-} \left(p_a^- \left[L_{S,a}^{0,-}(h) - L_{D,a}^{0,-}(h) \right] + \lambda K_{S,a}^- \left[L_{S,a}^-(h) - L_{D,a}^-(h) \right] \right) \\
& \qquad + \sum_{a=1}^{C} \sqrt{n} \left(\frac{n_a^+}{n} - p_a^+ \right) L_{S,a}^{0,+}(h) + \sum_{a=1}^{C} \sqrt{n} \left(\frac{n_a^-}{n} - p_a^- \right) L_{S,a}^{0,-}(h).
\end{aligned}
$$

Step 2. Since $\lim_{n \to +\infty} \frac{n_a^{\pm}}{n} = p_a^{\pm} > 0$ almost surely, this implies trivially that $\lim_{n \to +\infty} n_a^{\pm} = +\infty$ almost surely, for any a. Now, recall that, thanks to the continuous mapping theorem, it holds that for any $n_a^{\pm}$, $\lim_{n \to +\infty} \sqrt{n/n_a^{\pm}} = 1/\sqrt{p_a^{\pm}}$ almost surely. Finally, $\lim_{n \to +\infty} K_{S,a}^{\pm} = K_{D,a}^{\pm}$ almost surely, since $\nabla\phi$ is continuous and $\lim_{n \to +\infty} \mathbf{L}_S^{\pm}(h) = \mathbf{L}_D^{\pm}(h)$ almost surely.

Step 3. Now consider each term $a^\pm$ separately.

$$\begin{aligned}
&\sqrt{n_a^\pm}\left(p_a^\pm\left[L_{S,a}^{0,\pm}(h)-L_{D,a}^{0,\pm}(h)\right]+\lambda K_{S,a}^\pm\left[L_{S,a}^\pm(h)-L_{D,a}^\pm(h)\right]\right)\\
&\quad=\sqrt{n_a^\pm}\left(\left[p_a^\pm L_{S,a}^{0,\pm}(h)+\lambda K_{D,a}^\pm L_{S,a}^\pm(h)\right]-\left[p_a^\pm L_{D,a}^{0,\pm}(h)+\lambda K_{D,a}^\pm L_{D,a}^\pm(h)\right]\right)\\
&\qquad\qquad+\lambda\left(K_{S,a}^\pm-K_{D,a}^\pm\right)\sqrt{n_a^\pm}\left[L_{S,a}^\pm(h)-L_{D,a}^\pm(h)\right].
\end{aligned}$$

By the Central Limit Theorem (CLT), $\sqrt{n_a^\pm}\left[L_{S,a}^\pm(h)-L_{D,a}^\pm(h)\right]\xrightarrow[n\to+\infty]{d}N(0,[\sigma_a^\pm]^2)$, but since $\left(K_{S,a}^\pm-K_{D,a}^\pm\right)\xrightarrow[n\to+\infty]{a.s.}0$, the whole term $\left(K_{S,a}^\pm-K_{D,a}^\pm\right)\sqrt{n_a^\pm}\left[L_{S,a}^\pm(h)-L_{D,a}^\pm(h)\right]$ goes to 0 in probability (by Slustky's lemma).

All that remains is to apply the CLT to

$$\sqrt{n_a^\pm}\left(\left[p_a^\pm L_{S,a}^{0,\pm}(h)+\lambda K_{D,a}^\pm L_{S,a}^\pm(h)\right]-\left[p_a^\pm L_{D,a}^{0,\pm}(h)+\lambda K_{D,a}^\pm L_{D,a}^\pm(h)\right]\right),$$

and notice that $\mathbb{V}_{s=a,y=\pm1}\left[p_a^\pm\ell^{0,\pm}(h,z)+\lambda K_{D,a}^\pm\ell^\pm(h,z)\right]=[p_a^\pm]^2[\sigma_a^{0,\pm}]^2+\lambda^2[K_{D,a}^\pm]^2[\sigma_a^\pm]^2+2p_a^\pm\lambda K_{D,a}^\pm\mathrm{cov}_a^\pm\left(\ell^0(h,z),\ell^\pm(h,z)\right)$.

Since each data point in the sample is independent, the CLT can be applied term by term (i.e., all $a^\pm$'s have independent limiting distributions).

Step 4. To finish the proof, it simply remains to consider the adjustments due to observing empirical proportions rather than the true class probabilities. The limiting distribution of the vector $\sqrt{n}\left(\frac{n_a^\pm}{n}-p_a^\pm\right)_{\pm,a=1,\cdots,C}$ thus needs to be considered. However, this is simply the limiting distribution of empirical proportions in repeated trials of a multinoulli distribution, hence the result. □

The non-vanishing assumption on the gradient of the fairness loss function ϕ can be relaxed but leads to the need for a higher-order delta method (DasGupta, 2008).

4.6.2 Variance decomposition

The limiting variance derived in Proposition 5 is slightly cumbersome but can be expressed rather neatly as

$$\mathbb{V}_{\lim}[h]=\sum_{a^\pm}p_a^\pm\left(\sigma_a^{0,\pm}\right)^2+\lambda^2\frac{(K_a^\pm)^2}{p_a^\pm}\left(\sigma_a^\pm\right)^2+\text{covariance terms}\tag{4.18}$$

In a nutshell, the variance of the limiting trade-off distribution is the addition of

- The sum of intra-category $\ell^{0,\pm}$ losses weighted by their true probabilities $p_a^\pm$;

- The sum of intra-category fairness $\ell^\pm$ losses weighted by their *inverse* probabilities $p_a^\pm$, as in the uniform convergence framework. In addition, the constant $(K_a^\pm)^2$ represents the sensitivity of the fairness loss function ϕ and is bounded from above by $[K^\phi]^2$ in the case where ϕ is K^ϕ-Lipschitz continuous, while λ is the hyper-parameter governing the trade-off between statistical and fairness performances.

Qualitatively, the results obtained in the asymptotic and uniform frameworks are similar and point to the conclusion that to learn fairness, one must start by learning category samples. A low probability $p_a^{\pm}$ may not impact the variance of the overall statistical loss but may lead to a poor understanding of fairness trade-offs.

4.6.3 Numerical example

The behaviour of several debiasing methods, both in- and out-of-sample, has been studied on real and surrogate data in the literature ((Agrawal et al., 2020; Oneto et al., 2020; Zafar et al., 2017a) for instance), thus only an illustration of this result on a simple test case is provided, to highlight the role played by sampling variance in dealing with fairness metrics. In keeping with the previous section, suppose that a given classifier $\widehat{y} = h(\mathbf{x}, s)$ has been chosen, and consider its statistical properties on a (large) out-of-sample dataset.

4.6.3.1 Predictive parity and its limiting variance

In this simplified experimental setup, consider $C = 2$, and set $\ell^0 = 0$ with $\lambda = 1$, i.e., only look at a fairness criterion, and choose a ratio τ_{PP} linked to predictive parity (see Table 4.1):

$$\phi_D(h) = \tau_{D,\mathrm{PP}} := \frac{\mathbb{P}(\widehat{y} = 1|y = 1, s = 1)}{\mathbb{P}(\widehat{y} = 1|y = 1, s = 2)}. \tag{4.19}$$

The perfectly fair case corresponds to $\tau_{D,\mathrm{PP}} = 1$. The corresponding sample version of this criterion is also defined, using the samples $\mathscr{N}_a^+$, $a = 1, 2$:

$$\phi_S(h) = \tau_{S,\mathrm{PP}} = \frac{\frac{1}{n_1^+} \sum_{i \in \mathscr{N}_1^+} \mathbf{1}_{\{\widehat{y}_i = 1\}}}{\frac{1}{n_2^+} \sum_{i \in \mathscr{N}_2^+} \mathbf{1}_{\{\widehat{y}_i = 1\}}}. \tag{4.20}$$

In this case, $\mathbb{V}_{\mathrm{lim}}(h)$ can be computed directly as

$$\mathbb{V}_{\mathrm{lim}}(h) = \frac{1 - \mathbb{P}(\widehat{y} = 1|y = 1, s = 1)}{\mathbb{P}(\widehat{y} = 1|y = 1, s = 1)} \frac{1}{\pi_1} + \frac{1 - \mathbb{P}(\widehat{y} = 1|y = 1, s = 2)}{\mathbb{P}(\widehat{y} = 1|y = 1, s = 2)} \frac{1}{\pi_2}, \tag{4.21}$$

where $\pi_1 = \frac{p_1^+}{p_1^+ + p_2^+}$ and $\pi_2 = 1 - \pi_1 = \frac{p_2^+}{p_1^+ + p_2^+}$.

4.6.3.2 Simulation set-up and results

In simulations that were run, the fixed values $\mathbb{P}(\widehat{y} = 1|y = 1, s = 1) = 0.95$ and $\mathbb{P}(\widehat{y} = 1|y = 1, s = 2) = 0.99$ were used, leading to $\tau_{D,\mathrm{PP}} = 0.96$. The parameters $m = n_1^+ + n_2^+$, which represents the overall sample size available to us to compute $\tau_{S,\mathrm{PP}}$, and π_1, which is the effective proportion of class 1 versus class 2, were varied. Throughout this exercise, the experiment was simulated $N = 1000$ times. The baseline case is $m = 1{,}000$ and $\pi_1 = 10\%$.

From Figure 4.1 (a), one recovers the usual behaviour due to sample size: as m increases, the empirical distribution of $\tau_{S,\mathrm{PP}}$ becomes more peaked. A small sample size (such as $m = 100$) would yield poor accuracy and one can observe that the density is actually bi-modal.

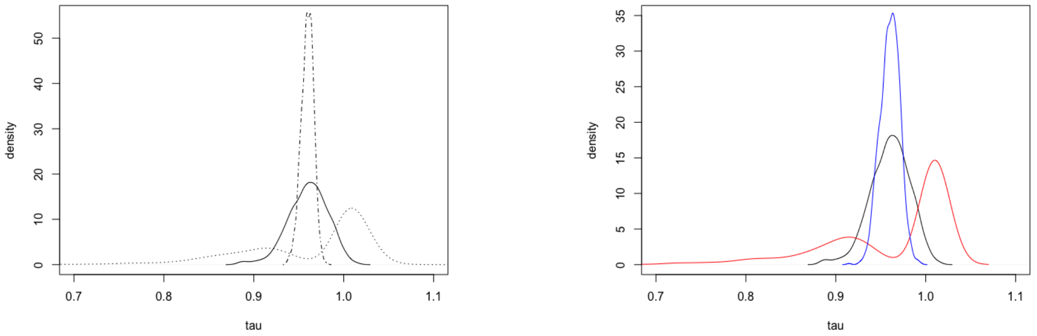

Figure 4.1: **(a)** Empirical densities of $\tau_{S,\text{PP}}$ for $m = 100$ (*dotted*), 1000 (*black*) and 10000 (*dashed*) at $\pi_1 = 10\%$. **(b)** Empirical densities of $\tau_{S,\text{PP}}$ for $\pi_1 = 1\%$ (*red*), 10% (*black*) and 50% (*blue*) at $m = 1000$.

The behaviour with respect to sample size can also be checked by comparing the empirical distribution of $\frac{\tau^{(k)}_{S,\text{PP}} - \tau_{D,\text{PP}}}{\sqrt{\mathbb{V}_{\text{lim}}(h)}}$, where $k = 1, \cdots, N$ is the experiment run, with a standard Gaussian distribution, as predicted by Proposition 5. It can be seen in Figure 4.2 that there is good adequacy between both distributions.

The most salient feature, however, is the behaviour of $\tau_{S,\text{PP}}$'s distribution with respect to category probability, as described in Figure 4.1 (b), where one can see that as the sample becomes balanced, the variance of the estimator $\tau_{S,\text{PP}}$ decreases. When π_1 is very small, say 1%, one recovers the bi-modal distribution that comes with a small sample.

4.6.3.3 Key observation

In a nutshell, low sample sizes and class imbalance can lead to poor generalisation. Indeed, by considering the cases $m = 100, \pi_1 = 10\%$ and $m = 1000, \pi_1 = 1\%$, it is clear that one cannot draw any conclusion regarding the presence of bias nor in terms of which group is the advantaged group. Bi-modality is a particularly interesting characteristic of the empirical distribution: there is a mode strictly below 1 and a mode strictly above 1.

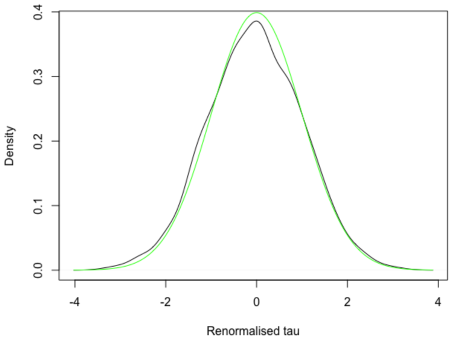

Figure 4.2: Normalised density of $\tau_{D,\text{PP}}$ (in *solid black*) and theoretical limiting distribution (in *green*) for $m = 1{,}000$ and $\pi_1 = 10\%$.

4.7 Experimental results

This Section investigates the theoretical results' validity on real datasets. Here, the task to be solved (via the non-linear L-BFGS solver) is a logistic regression with an additional constraint on fairness. Concretely, the objective function is defined as:

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i; \beta) + \lambda \left[\frac{1}{n_0} \sum_{j=1}^{n_0} \ell(y_j, \mathbf{x}_j; \beta) - \frac{1}{n_1} \sum_{j'=1}^{n_1} \ell(y_{j'}, \mathbf{x}_{j'}; \beta) \right]^2, \tag{4.22}$$

where ℓ is the logistic loss function and the groups 0 and 1 correspond to two different classes of a protected attribute. Three datasets with binary protected attributes and binary outcomes were considered, shown in Table 4.2, and a standard 80–20 train-test split was applied.

Table 4.2: Description of datasets used in the proposed experiments.

Dataset name	Reference	Protected attribute	Binary outcome
Adult	(Dua & Graff, 2017)	Sex	Income exceeds 50$K/year
Loan defaults	(Yeh & hui Lien, 2009)	Sex	Default payment next month
German credit	(Dua & Graff, 2017)	Sex	Credit-worthiness

Figures 4.3 and 4.4 illustrate the trade-off between the average loss (i.e., $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i; \beta)$) and disparity (i.e., $\left| \frac{1}{n_0}\sum_{j=1}^{n_0} \ell(y_j, \mathbf{x}_j; \beta) - \frac{1}{n_1}\sum_{j'=1}^{n_1} \ell(y_{j'}, \mathbf{x}_{j'}; \beta) \right|$) for a range of λ values. [3] On the *train test* (Figure 4.3, a clear dependence between the value of λ and the respective average loss and disparity between groups is visible. As expected, an increase in λ tends to decrease disparity while increasing the average loss, and vice versa.

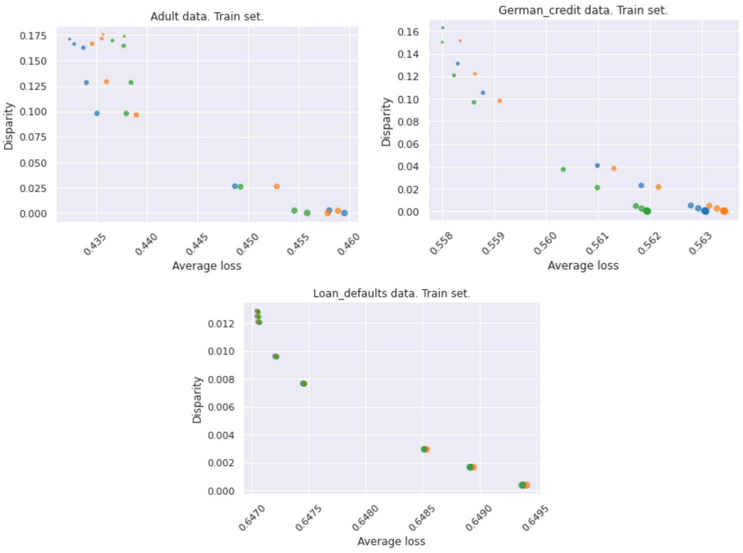

Figure 4.3: The trade-off between disparity and average loss on the Adult, German Credit, and Loan Defaults datasets. Each colour corresponds to a different initialisation, while the symbol's size corresponds to the weight λ on the disparity in the overall loss function. Results are measured on the *train set*.

[3] Specifically $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500\}$. Each distinct colour represents a different initialisation, while the dot size is a monotonically increasing function of the respective λ.

However, this is not always the case on the *test set* (Figure 4.4), as some intermediate values of λ achieve both lower average loss and lower disparity, whereas more (or totally) debiased models clearly overfit.

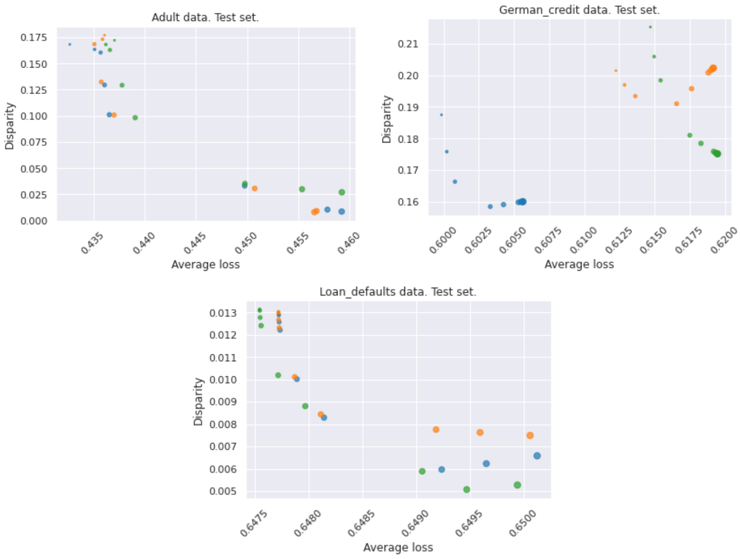

Figure 4.4: The trade-off between disparity and average loss on the Adult, German credit, and Loan defaults datasets. Each colour corresponds to a different initialisation, while the symbol's size corresponds to the weight λ on the disparity in the overall loss function. Results are measured on the *test set*.

4.7.1 Standard deviation scaling

It is possible to experiment with class imbalance to see how the solution to the original problem changes as imbalance grows, in line with Theorem 5. To do so, observations from one class are down-sampled, keeping the original number of observations in the other. As predicted, on the test set, the loss variance increases with class imbalance, as shown in Figure 4.5[4].

4.8 Conclusion

The analysis conducted from two different angles, namely learning with uniform convergence and asymptotics, leads to the same overall qualitative assessment. While probabilistic upper bounds were derived to prove learnability guarantees on the one hand and convergence to a limiting distribution on the other, a striking feature of learning fairness trade-offs is the fundamental difference of regimes between the usual statistical performance criterion measured on the whole sample and the fairness penalty that examines relationships between sub-samples.

Indeed, fairness is not about learning an average distribution, quite the contrary, in the sense that it requires a fine understanding of differences across conditional distributions. Intuitively, in the usual case of an overall statistical loss function, if a category only represents a small portion of the sample,

[4] Here, points of the same colour correspond to the same random seed. Different random seeds are needed to obtain several variants of the down-sampled data. The dotted colour lines are the plain linear least square estimate through the points of the same colour, while the bold black line is the least squares' estimate using data points from all seeds. λ is fixed at 50.

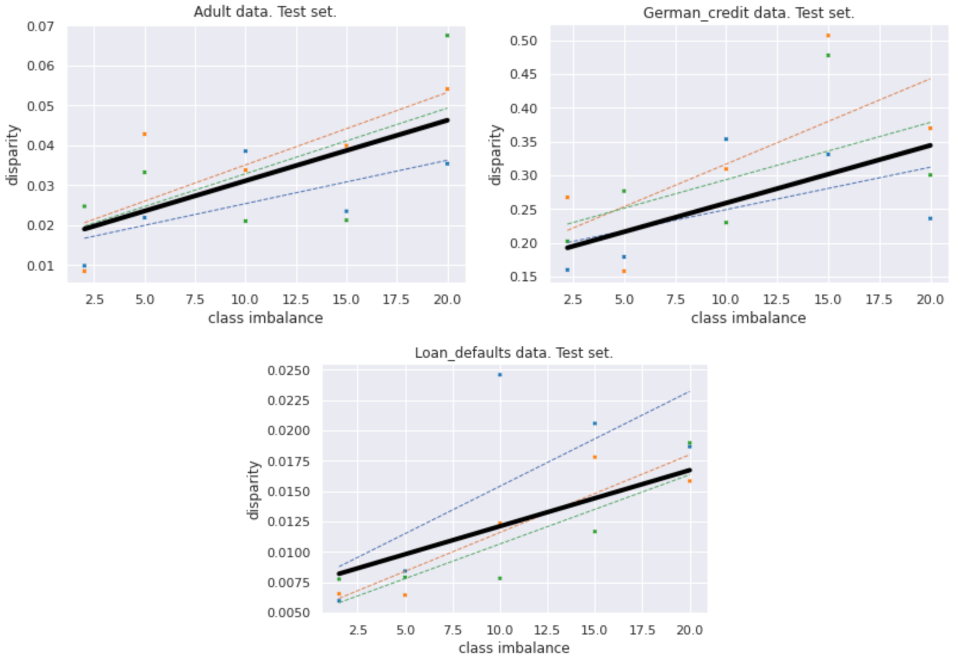

Figure 4.5: Fairness uncertainty versus class imbalance ratio in protected characteristic on Adult, German credit, and Loan defaults datasets. Results are measured on the test set.

it also constitutes only a small fraction of the overall loss. On the contrary, from a fairness viewpoint, it is difficult to make any statement regarding the relationship between this particular category and other categories, whence a high risk for a given fairness objective.

With a simple numerical example, it has also been shown that these theoretical considerations translate to practical cases and that small sample sizes or class imbalances could lead to spurious empirical results. In particular, the presence, under these conditions, of bi-modality means that performing debiasing –whether total or partial– may actually be counterproductive.

There are, however, multiple avenues for learning fairness trade-offs involving better tuning of hyper-parameters (such as λ in our case) of debiasing algorithms or identifying underlying causal connections linking protected classes to features. But another research direction seems to be the design of fairness functions that account for the shortcomings of having possibly lower sample sizes. One possibility is to optimise fairness trade-offs with data-dependent upper bounds, such as those we provided in a P.A.C. framework. In any case, sample-efficient techniques for bias mitigation are not a nice-to-have but a must-have.

Chapter 5

Fair Generalised Low-Rank Models

This chapter is the result of a conference paper co-authored with Islam Utyagulov. F.B.G. conceived of the presented ideas, developed the theoretical aspects, designed the experiments, and wrote the manuscript. I.U. contributed to the code base and ran the numerical experiments. Both authors discussed the results and commented on the manuscript.

Research objectives

Data representation, via visualisation or dimensionality reduction, is key in framing problems or as a first step in answering them. Ensuring that such representations are fair is thus paramount. Bias-mitigating techniques are now well established in the supervised learning literature and have shown their ability to tackle fairness-accuracy and fairness-fairness trade-offs. These are usually predicated on different conceptions of fairness, such as demographic parity or equal odds that depend on the available labels in the dataset. However, it is often the case in real life that unsupervised learning is used as part of a machine learning pipeline (for instance, to perform dimensionality reduction or representation learning via singular value decomposition) or as a standalone model (for example, to derive a customer segmentation via k-means). It is thus crucial to develop approaches towards fair unsupervised learning. This work investigates fair unsupervised learning within the broad framework of generalised low-rank models (GLRM). Importantly, the concept of fairness functional is introduced, encompassing both traditional unsupervised learning techniques and min-max algorithms (whereby one minimises the maximum group loss).

5.1 Introduction

Using unsupervised learning algorithms, such as PCA, k-means or non-negative matrix factorisation – which is prevalent in recommender systems – without paying attention to fairness may lead to adverse outcomes in some particular demographic groups (e.g., customer segmentation or facial recognition). Indeed, one verifies on a number of datasets that there are discrepancies, sometimes important, between the average cost (such as reconstruction error or distance to a centroid) born by one group versus another. Thus, the fairness metric that emerges in these applications is that of (average) cost parity amongst groups.

More broadly, recent advances have been made recently in the space of fair unsupervised learning, in particular by introducing fairlets (Chierichetti et al., 2017), leading to fair PCA (Samadi et al., 2018; Tantipongpipat et al., 2019), fair k-medoids (Ahmadian et al., 2019; Ghadiri et al., 2021; Kleindessner et al., 2019a) and fair spectral clustering (Kleindessner et al., 2019b). Since fairness in this context is not obvious to tackle, there have been multiple attempts to define it. A recent overview of fair clustering has been given in (Chhabra et al., 2021). On the other hand, in supervised learning, multiple technical definitions of fairness co-exist and have been reviewed in-depth (Berk et al., 2018; Kim et al., 2020b; Narayanan, 2018; Verma & Rubin, 2018), bringing to the fore impossibility theorems (Chouldechova, 2017a; Kleinberg et al., 2017) that proved that these different acceptations of fairness cannot be satisfied at once. In addition, it has been shown that (Agrawal et al., 2020; Kim et al., 2020b; Pleiss et al., 2017) fully debiased models could fail to generalise out-of-sample, which we would expect to also apply to the case of unsupervised learning.

Contributions. The main result is a framework that encompasses many applications such as fair PCA (Samadi et al., 2018), fair k-medoids (Ghadiri et al., 2021), fair non-negative matrix factorisation and other models whose standard versions can be expressed as generalised low-rank models (Udell et al., 2016), and provides added flexibility.

- First, a general fair generalised low-rank framework is developed, which reduces disparity across the group-wise average cost in an unsupervised learning task (such as reconstruction error in PCA).

- Second, it is shown that a particular group functional, namely weighted Log-Sum Exponential, has interesting properties (such as convexity, differentiability, etc.) that make it particularly appropriate.

- Third, buidling on (Udell et al., 2016), a generic algorithm is designed, taking advantage of *biconvexity*. This generality in specifying a fair GLRM model makes this framework a very flexible one, also including *partial debiasing*.

- Fourth, the proposed methodology is applied to multiple datasets, being benchmarked against fair PCA and fair k-means algorithms, and its performance is shown out-of-sample. This highlights the role of partial debiasing.

Note that a number of extensions are possible by considering relative costs or outcome-based fairness.

5.2 Generalised Low-Rank Models

Some concepts linked to generalised low-rank models ("GLRMs") are first recalled. A textbook exposition of GLRMs is given in (Udell et al., 2016). The term itself, *generalised low-rank models* refers –in general– to the approximation of a data matrix by the product of two low-dimensional factors.

Definition 9. A generalised low-rank model is defined based on the following elements:

1. An $n \times p$ data matrix A and a (lower) dimension $d < p$;

2. Element-wise (usually biconvex) loss functions $\ell_{i,j}$, (usually convex) penalty functions r_i and $\tilde{r}_j$ for all $i = 1, \cdots, n$ and $j = 1, \cdots, p$, and an overall loss function $\tilde{\mathscr{L}}$ written as

$$\tilde{\mathscr{L}}(X,Y) = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \ell_{i,j}(x_i \cdot y_j, A_{i,j}) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j), \tag{5.1}$$

   where $x_i \in \mathbb{R}^d$ and $y_j \in \mathbb{R}^d$ for all i, j. The matrices X and Y correspond to the stacked vectors (i.e., the row-wise concatenation of the x_i's and the y_j's respectively).

Remark 12. In this work, the matrix Y is understood to be made up of the row entries y_j ($j = 1, \cdots, p$) as the *dictionary* to be learnt and the matrix X made up of the row entries x_i ($i = 1, \cdots, n$) as the individual *weights*.

Some examples can now be provided:

1. By choosing $\ell_{i,j}(u,a) = (u-a)^2$, one recovers PCA.

2. Similarly, robust PCA can be obtained by picking $\ell_{i,j}(u,a) = |u-a|$ with $r(x) = \gamma/2\|x\|_2^2$ and $\tilde{r}(y) = \gamma/2\|y\|_2^2$.

3. On the other hand, setting $r(x) = 0$ if $x \geq 0$ and $+\infty$ otherwise, with the same definition for $\tilde{r}$ leads to non-negative matrix factorisation ("NNMF").

4. Finally, picking $r(x) = 0$ if $x = e_l$ for some $l \in \{1, \cdots d\}$ and $+\infty$ otherwise, while $\tilde{r}(y) = 0$, leads to the usual k-means clustering problem.

Many more applications (such as subspace clustering) can be shown to fit the generic GLRM form (Udell et al., 2016).

5.3 Fairness in unsupervised learning

5.3.1 Background

The relationship between fairness in supervised and unsupervised learning is not straightforward. Many notions of fairness in supervised learning (such as classification and scoring (Chouldechova, 2017b; Hardt et al., 2016; Kleinberg et al., 2017)) focus on a single learning task, whereas unsupervised learning considers a generic transformation of the data. While fairness in unsupervised learning has recently become a major theme of research, fair PCA –for instance– can be seen directly in the line of earlier attempts to reduce the correlation between a protected (or sensitive) attribute (Calmon et al., 2017; Zemel et al., 2013).

For the sake of brevity, an exhaustive account of fairness in unsupervised learning is not provided here. It is tempting to consider (McLachlan & Basford, 1988) as an early attempt at introducing (individual) fairness in clustering. By adapting the notion of disparate impact to clustering and introducing the notion of fairlets (i.e., minimal sets that satisfy fair representation while approximately preserving the clustering objective), (Chierichetti et al., 2017) paved the way for much of the work in the field. (Samadi et al., 2018; Tantipongpipat et al., 2019) explore PCA and dimensionality reduction with multiple constraints, with an application to fairness. In (Kleindessner et al., 2019b), the authors tackle the case of spectral clustering. (Abbasi et al., 2021) proposes a generic approach to fair clustering, including k-medoids and considers a min-max criterion across groups. On a different note, (Celis et al., 2018) tackles the issue of data summarisation via a determinantal measure of diversity.

Furthermore, work in fair recommender systems (which include some matrix factorisation techniques) has grown due to a better understanding of certain phenomena, such as echo chambers or filter bubbles. In addition to biases linked to specific protected characteristics (such as poor performance of recommender systems to serve under-represented minority groups), specific issues have appeared, such as *user under-representation* (Li et al., 2021) and *item under-recommendation* (also known as popularity bias) (Zhu et al., 2020).

5.3.2 Fairness criteria

As a result of unsupervised learning's diversity and breadth, multiple notions of fairness have been put forward for (or adapted to) unsupervised learning, including *social fairness*, *balance fairness* and *individual fairness*. A brief account of these different criteria is indicated here.

1. The *social fairness* criterion was introduced in unsupervised learning in (Ghadiri et al., 2021), where it was applied to a clustering problem and further developed in (Abbasi et al., 2021; Makarychev & Vakilian, 2021). In short, it requires that the average cost (e.g., reconstruction loss in PCA or distance to medoid in clustering) be the same across groups. This can be tackled by minimising the maximum of the groups' average costs. Note that it has had a long history since it was introduced by philosopher John Rawls in his *Theory of Justice* (Rawls, 1971) as a justice criterion ("maximin") applied to the usual utilitarian framework.

2. Similarly, the principle according to which different groups should have the same distribution across clusters can be traced back to (Chierichetti et al., 2017), where the authors posit the notion of *balance fairness*, which they attack through so-called "fairlets". A related concept is that of *bounded representation* (Ahmadian et al., 2019), which requires that the proportion of a group in each cluster be between two pre-specified values. The *maximum fairness cost*

(Ghadiri et al., 2021) is the maximum of the sum of all deviations from the ideal proportion for each protected group in a cluster.

3. Last, *individual fairness* compares the statistical distance between two points obtained from their inputs and the algorithm's output distribution and mandates that two similar inputs should have similar outputs. This paradigm was first used in (McLachlan & Basford, 1988) for clustering and further adapted by (Kleindessner et al., 2020).

Many more concepts exist (Chhabra et al., 2021) (the reader is also referred to (Narayanan, 2018) for an overview of such concepts in supervised learning) but this paper focuses primarily on *social fairness*-type of metrics. Note, however, that –as pointed out in Section 5.6– the proposed framework can be easily adapted to other fairness notions.

5.4 Group functionals

In this Section, the key insight of this chapter's proposal is introduced, namely that of a group functional. Suppose that there are K (distinct) groups, $k = 1, \cdots, K$, corresponding, say, to the K categories of the protected characteristic s. The corresponding partition of Ω can thus be introduced:

$$\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_K, \tag{5.2}$$

such that the intersection between any Ω_k and $\Omega_{k'}$ is empty.

5.4.1 Motivation

It is immediate to notice that Equation 5.1 is simply

$$\mathscr{L}(X,Y) = \sum_{k=1}^{K} \frac{|\Omega_k|}{|\Omega|} z_k(X,Y) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j), \tag{5.3}$$

where the following shorthand notation is used

$$z_k(X,Y) := \frac{\sum_{(i,j)\in\Omega_k} \ell_{i,j}(x_i y_j, A_{i,j})}{|\Omega_k|},$$

which can be rewritten as $\mathscr{L}(X,Y) = T(z_1(X,Y), \cdots, z_K(X,Y)) + \text{Penalty terms}$, and $T(z_1, \cdots, z_K) = \sum_{k=1}^{K} \frac{|\Omega_k|}{|\Omega|} z_k$. In particular, T is simply a linear combination of the average cost in each group weighted by this group's proportion in the sample.

Fair GLRMs thus consist of finding functionals that are more suited to the task of reducing disparities. Finally, note that, by definition, this setup applies to any model that can be expressed as a generalised low-rank model, and thus encompasses not only PCA and k-means but also sparse PCA (sPCA), non-negative matrix factorisation (NMF), etc.

5.4.2 Fair GLRMs

Recalling the assumptions defining generalised low-rank models leads to the definition of *fair* GLRMs by modifying the loss function:

Definition 10. Suppose that $T : \mathbb{R}^K \to \mathbb{R}$ is a non-decreasing function in each of its arguments, then the fair generalised low-rank model with respect to T is defined as minimising the following objective function:

$$\mathscr{L}(X,Y) = T\left(z_1(X,Y), \cdots, z_K(X,Y)\right) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j). \tag{5.4}$$

While this definition can apply to all functional T, the ones we consider aim at reducing disparities across groups. Presenting some concrete cases is now in order.

Remark 13. Note that some extreme cases are possible:

1. If $K = 1$ (such that only one group is present), then the fGLRM is equivalent (up to a non-decreasing transformation) to the standard GLRM one.

2. If $K = |i \in \Omega| = n$ (i.e., each row of the data matrix A is a group), then the fairness functional ensures that the maximal *individual* cost is minimised.

5.4.3 Examples

Some well-known fairness functionals can be introduced, mostly stemming from the supervised learning literature.

Standard and reweighed loss function Setting $T(z_1, \cdots, z_K) = \sum_{k=1}^{K} w_k z_k$, where $w_k \geq 0$ ad $\sum_{k=1}^{K} w_k = 1$. When $w_k = |\Omega_k|/|\Omega|$, the usual generalised low-rank model is recovered. If, on the other hand, $w_k = \frac{1}{K}$, then this corresponds to the *reweighed* GLRM. This is akin to some pre-processing techniques used in (Kamiran & Calders, 2012) for example.

Min-Max Choosing $T(z_1, \cdots, z_K) = \max_{k=1,\cdots,K} z_k = \|\mathbf{z}\|_\infty$ leads to a min-max problem. Note that this is the functional implicitly chosen in (Samadi et al., 2018) (cf. Lemma 4.8 and Proof of Theorem 4.5 therein) to tackle fair PCA and in (Ghadiri et al., 2021) to handle k-means. The min-max approach has the intuitive justification

(Weighted) L^p norms A possible choice is a weighted L^p norm $T(z_1, \cdots, z_K) = \left(\sum_{k=1}^{K} w_k z_k^p\right)^{\frac{1}{p}} = \|\mathbf{z}\|_{p,\mathbf{w}}$. If the weights are uniform, then it follows that $T(z_1, \cdots, z_K) \propto \|\mathbf{z}\|_p$, which is –up to a simple transformation– a setup used in (Li et al., 2020b). In particular, since the limit of the $\|\cdot\|_p$ norm is the $\|\cdot\|_\infty$ norm as $p \to +\infty$, one recovers the min-max formulation as an extreme case.

Penalised learning Adding a term penalising unfairness and disparities is fairly common to (partially) debias supervised learning algorithms (Donini et al., 2018; US Congress, 2003) and leads to fairness functionals of the type $T(z_1, \cdots, z_K) = \sum_{k=1}^{K} w_k z_k + \lambda \sum_{k,k'} d\left(z_k, z_{k'}\right)$, where d is a chosen distance (such as L^1 or L^2) and $\lambda > 0$ tunes the trade-off between the statistical loss and the disparity penalty term. One may avoid the double sum in the penalty term by considering instead $\sum_{k=1}^{K} w_k z_k + \lambda \sum_{k'} d\left(z_{k'}, \sum_{k=1}^{K} w_k z_k\right)$.

Building new fairness functionals from old ones From J existing fairness functionals $T_1, \cdots, T_J$, one can create a new functional V

$$V(z_1, \cdots, z_K) = \mathscr{M}\left(T_1(z_1, \cdots, z_K) \cdots, T_J(z_1, \cdots, z_K)\right), \tag{5.5}$$

where $\mathscr{M} : \mathbb{R}^J \mapsto \mathbb{R}$ is a function that is non-decreasing in each of its components. The most straightforward example is to pick a convex combination of fairness functionals

$$V(z_1, \cdots, z_K) = \sum_{j=1}^{J} \lambda_j T_j(z_1, \cdots, z_K),$$

where $\lambda_j \geq 0$ and $\sum_{j=1}^{J} \lambda_j = 1$. For instance, one may consider functions V that "interpolate" between the usual average loss and the min-max case, as one may wish to control the fairness-accuracy control. For instance, one can pick $\gamma \in (0,1)$ such that

$$V(z_1, \cdots, z_K) = \lambda \left(\sum_{k=1}^{K} w_k z_k\right) + (1-\lambda)\max(z_1, \cdots, z_K) \tag{5.6}$$

In this work, a very specific group functional is used (but most considerations apply to any T).

5.4.4 Bayesian interpretation

The traditional GLRM framework offers a natural Bayesian interpretation, following (Fithian & Mazumder, 2018). Indeed, the minimisation of an fGLRM objective in Equation 5.4 can be seen as a *maximum a posteriori* problem, such that the hierarchical Bayesian model reads:

$$e^{-T(z_1(X,Y), \cdots, z_K(X,Y))} \cdot \prod_{i=1}^{n} e^{-r_i(x_i)} \cdot \prod_{j=1}^{p} e^{-\tilde{r}_j(y_j)}, \tag{5.7}$$

where the prior distributions on x_i and y_j have probability density functions proportional to $e^{-r_i(x_i)}$ and $e^{-\tilde{r}_j(y_j)}$ respectively. In the case of $T(z_1, \cdots, z_K) = \sum_{k=1}^{K} w_k z_k$, we recover

$$e^{-T(z_1(X,Y), \cdots, z_K(X,Y))} = \prod_{(i,j) \in \Omega} e^{-\frac{w_{k(i)}}{|\Omega_{k(i)}|} \ell_{i,j}(x_i \cdot y_j; A_{ij})}.$$

In short, in this example, each entry $A_{i,j}$ is taken to be independent, but not necessarily identically distributed, depending on the values of the ratio $\frac{w_{k(i)}}{|\Omega_{k(i)}|}$.

However, generally, the product structure is *not* preserved and the group functional T introduces some dependence across observations, so that the observations A_{ij} are not independent anymore. Similarly, if one adopts the outcome-based version of fGLRMs in Equation A.13, then the prior distributions are not independent either. To summarise, fair GLRMs induce a *dependent* hierarchical

Bayesian model.

5.4.5 Log-sum exponential functional

Throughout this work, the main fairness functional in use is the weighted Log-Sum Exponential (LSE) due to its many desirable properties and its ability to interpolate between the standard GLRM and the min-max programme.

5.4.5.1 Defining *wLSE*

The weighted (scaled) Log-Sum Exponential is introduced. It is a (small) generalisation of the usual Log-Sum Exponential, which is widely used in other machine learning applications (see (Nielsen & Sun, 2016)).

Definition 11. The weighted Log-Sum-Exponential ("wLSE") is defined as

$$T(z_1, \cdots, z_K) = \frac{1}{\alpha} \log \left(\sum_{k=1}^{K} w_k \, e^{\alpha z_k} \right), \tag{5.8}$$

where $\alpha > 0$, $w_k \geq 0$ for all k and $\sum_{k=1}^{K} w_k = 1$.

Remark 14. First, notice that the weight normalisation requirement is not strictly necessary but useful. Second, one can pick the natural choice $w_k = \frac{|\Omega_k|}{|\Omega|}$, but can also perform some sample reweighing simultaneously.

5.4.5.2 Properties

The wLSE has many properties of interest (both theoretically and practically).

Proposition 19. *Suppose that $\alpha > 0$, $w_k \geq 0$ for all k and $\sum_{k=1}^{K} w_k = 1$, then the weighted Log-Sum Exponential verifies the following properties:*

1. *T is (jointly) convex in $(z_1, \cdots, z_K)$.*

2. *The weighted average and the maximum functions are recovered as limiting cases:*

$$\lim_{\alpha \to 0} T(z_1, \cdots, z_K) = \sum_{k=1}^{K} w_k z_k \tag{5.9}$$

$$\lim_{\alpha \to +\infty} T(z_1, \cdots, z_K) = \max(z_1, \cdots, z_K) \tag{5.10}$$

3. *A shift property holds for every $\bar{z} \in \mathbb{R}$: $T(z_1, \cdots, z_K) = \bar{z} + T(z_1 - \bar{z}, \cdots, z_K - \bar{z})$.*

Proof. See Appendix A.2.1. □

Remark 15. The shift property has a very natural explanation when thinking about fairness, as it decomposes the objective into the usual average cost, $\bar{z}$, (possibly reweighed), and a term that penalises disparity $T(z_1 - \bar{z}, \cdots, z_K - \bar{z})$. Based on this insight, one could tune the objective further: $T_\gamma(z_1, \cdots, z_K) = \bar{z} + \gamma T(z_1 - \bar{z}, \cdots, z_K - \bar{z})$, for $\gamma > 0$.

5.4.5.3 Choosing α and fairness implications

The hyper-parameter α enables one to "interpolate" between the traditional GLRM problem and its fair min-max formulation. Given that a number of articles use min-max formulations, it is worth justifying why one may wish to choose $\alpha \neq +\infty$. Indeed, $\alpha \neq +\infty$ introduces *partial debiasing* in unsupervised learning and helps relax assumptions of strict equal average costs amongst groups.

1. wLSE is a soft maximum and enables modellers to approximate the maximum with a differentiable function (Nielsen & Sun, 2016), which is an advantage in many circumstances, including when gradient descent-type algorithms are used to minimise the objective function.

2. Constraints may exist in the application of an algorithm, such as a minimum overall statistical performance, leading a modeller to debias an algorithm as much as possible while keeping the overall average loss below a given threshold.

3. Partial debiasing was used in (Kim et al., 2020b) to account for the presence of fairness-accuracy (or even fairness-fairness) trade-offs (Kleinberg et al., 2017; Narayanan, 2018). The notion of a trade-off between an average statistical performance metric (such as an empirical average loss) and disparity metrics is illustrated empirically in Section 5.7.

4. Issues regarding the out-of-sample performance of debiasing algorithms have been investigated (Agrawal et al., 2020; Chuang & Mroueh, 2021) in the context of supervised learning. (Agrawal et al., 2020), in particular, demonstrates the need to carefully tune a debiasing algorithm as "total" debiasing may lead to worse results out-of-sample. In other words, picking an intermediate value of α may lead to superior results on unseen data (such as the out-of-sample test set). Finally, it may not be optimal from a fairness point of view to choose $\alpha = +\infty$.

5.5 Fitting fair GLRMS

The focus in this Section is now on the minimisation of the objective function in Equation 5.4. At first glance, it may seem significantly more complex than in the case of standard GLRMs, but it turns out that –in most cases– the essential *biconvex* property of the objective function still holds.

5.5.1 Biconvexity of fGLRMs

The attractiveness of fGLRMs comes from the fact that under mild assumptions on the fairness functional T, they are biconvex functions in X and Y and thus fairly straightforward to minimise. Note that one cannot hope, in general, for something better than biconvexity since standard GLRMs are themselves biconvex.

Proposition 20. *Under the assumptions that $T : \mathbb{R}^K \to \mathbb{R}$ is convex and is non-decreasing in each argument, and each individual loss function $\ell_{i,j}$ is biconvex in x_i and y_j, then the application*

$$(X,Y) \mapsto T\left(z_1(X,Y), \cdots, z_K(X,Y)\right) \tag{5.11}$$

is biconvex in X and Y. If, in addition, each penalty function r_i or $\tilde{r}_j$ is convex, then the application

$$(X,Y) \mapsto \mathscr{L}(X,Y) = T\left(z_1(X,Y),\cdots,z_K(X,Y)\right) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j) \tag{5.12}$$

is biconvex in X and Y.

Proof. See Appendix A.2.2. □

What this result shows is that the introduction of a group functional does not change the fundamental structure of a generalised low-rank model. This has implications in terms of *optimisation*, as existing algorithms can simply be tweaked and reused. While one may use gradient descent algorithms and variants thereof on the non-convex objective function (Equation 5.4), more bespoke algorithms exist.

5.5.2 Alternating minimisation (or alternate convex search)

Alternating minimisation is a well-known algorithm that minimises the objective function in one direction at a time. If the objective function is multi-convex (i.e., convex in each direction when the other ones are fixed), this is the same as *alternate convex search*. The reader is referred to (Boyd & Vandenberghe, 2004; Gorski et al., 2007; Hastie et al., 2015b).

Algorithm 1 Alternating Minimisation for fGLRM A.3.1

```
Require: Matrix $\mathbf{A}$, loss functions $\ell_{i,j}$ and penalty functions $r_i$ and $\tilde{r}_j$.
  Select initial values $\mathbf{X}^0$ and $\mathbf{Y}^0$
  repeat
    for $i = 1,\cdots,n$ do
      $x_i \leftarrow \arg\min_x \mathscr{L}((\mathbf{X}_{-i},x),\mathbf{Y}) + r_i(x)$
    end for
    for $j = 1,\cdots,p$ do
      $y_j \leftarrow \arg\min_y \mathscr{L}(\mathbf{X},(\mathbf{Y}_{-j},y)) + \tilde{r}_j(y)$
    end for
  until convergence
  return $\mathbf{X},\mathbf{Y}$
```

The shorthand $\mathbf{X} = (\mathbf{X}_{-i},x_i)$ for all $i = 1,\cdots,n$ has been used. This algorithm is the adaptation to the fair set-up of Algorithm 1 in (Udell et al., 2016).

Remark 16. A couple of practical remarks can be made at this stage.

1. The for loop $i = 1,\cdots,n$ in this algorithm may be replaced with a standard GLRM for loop if the penalty function r_i is a set indicator penalty (for instance in the case of clustering, $r_i(x) = 0$ if $x = e_l$ for some $l \in \{1,\cdots d\}$ and $+\infty$ otherwise).

2. Due to overflow, it may sometimes be necessary to express the fairness functional slightly differently. For instance, $||\mathbf{z}||_{p,\mathbf{w}} = ||\mathbf{z}||_\infty \left|\left|\frac{\mathbf{z}}{||\mathbf{z}||_\infty}\right|\right|_{p,\mathbf{w}}$. Similarly, $wLSE_\alpha(\mathbf{z}) = ||\mathbf{z}||_\infty + wLSE_\alpha(z_1 - ||\mathbf{z}||_\infty,\cdots z_K - ||\mathbf{z}||_\infty)$.

5.5.3 Biconvex gradient descent

Suppose here that all functions are differentiable and that the penalty functions r_i and $\tilde{r}_j$ are convex. Then, thanks to the biconvexity of $\mathscr{L}$ in X and Y, one can derive the following expressions:

$$\begin{aligned}\frac{\partial \mathscr{L}(X,Y)}{\partial x_i} &= \frac{\partial T}{\partial z_{k(i)}}\frac{\partial z_{k(i)}}{\partial x_i} + \nabla r_i(x_i)\\ \frac{\partial \mathscr{L}(X,Y)}{\partial y_j} &= \sum_{k=1}^{K}\frac{\partial T}{\partial z_k}\frac{\partial z_k}{\partial y_j} + \nabla \tilde{r}_j(y_j)\end{aligned}$$

where $\frac{\partial z_{k(i)}}{\partial x_i} = \frac{1}{|\Omega_{k(i)}|}\sum_{j|(i,j)\in\Omega_{k(i)}} \nabla \ell_{i,j}(x_i \cdot y_j; A_{ij})y_j$, $\frac{\partial z_k}{\partial y_j} = \frac{1}{|\Omega_k|}\left(\sum_{i|(i,j)\in\Omega_k} \nabla \ell_{i,j}(x_i \cdot y_j; A_{ij})x_i\right)$, and $\frac{\partial T}{\partial z_k} = \frac{w_k e^{\alpha z_k}}{\sum_{k=1}^{K} w_k e^{\alpha z_k}}$. This leads to a biconvex gradient descent algorithm:

Algorithm 2 Biconvex Gradient Descent A.3.3

```
Require: Matrix $\mathbf{A}$, loss functions $\ell_{i,j}$ and penalty functions $r_i$ and $\tilde{r}_j$, step sizes $(\alpha_t)_{t\geq 1}$.
  Select initial values $\mathbf{X}^0$ and $\mathbf{Y}^0$
  $t \leftarrow 1$
  repeat
    for $i = 1, \cdots, n$ do
      $g_i^t \leftarrow \frac{\partial \mathscr{L}(\mathbf{X}^{t-1}, \mathbf{Y}^{t-1})}{\partial x_i^{t-1}}$
      $x_i^t \leftarrow x_i^{t-1} - \alpha_t g_i^t$
    end for
    for $j = 1, \cdots, p$ do
      $\tilde{g}_j^t \leftarrow \frac{\partial \mathscr{L}(\mathbf{X}^{t}, \mathbf{Y}^{t-1})}{\partial y_j^{t-1}}$
      $y_j^t \leftarrow y_j^{t-1} - \alpha_t \tilde{g}_j^t$
    end for
    $t \leftarrow t + 1$
  until convergence
  return $\mathbf{X}^t, \mathbf{Y}^t$
```

The main difference between GLRMs and fGLRMS comes from their particular gradient structure and the fact that an iterative weighing scheme has *implicitly* been introduced, similarly to boosting. Indeed, by denoting

$$\delta_k = \frac{\partial T}{\partial z_k} = \frac{w_k e^{\alpha z_k}}{\sum_{k=1}^{K} w_k e^{\alpha z_k}},$$

it follows that $\delta_k \geq 0$ for all k's and $\sum_{k=1}^{K}\delta_k = 1$. When $\alpha \to 0$, $\delta_k = w_k$ is recovered, and δ_k does not change at each iteration. On the other hand, when α is non-zero, the weights are adaptive and over-weigh the groups with higher average costs in the previous iteration.

Remark 17. Some convergence properties of these algorithms are discussed in the Appendix A.3.

5.6 Outcome-based fairness and other Extensions to fGLRMs

In this Section, it is shown how alternative notions of fairness can be included in the fGLRM framework. Importantly, the alternating minimisation algorithm can still be applied to these cases.

5.6.1 Group functional on outcome disparity

Suppose that one adopts an outcome-based viewpoint on unsupervised learning by considering, for instance, that one wishes to apply notions of demographic parity to the output of the unsupervised learning algorithm, which is considered here to be $x_i \cdot y_j$. In recommender systems, for example, one may wish to ensure that all groups have the same (predicted) average rating or satisfaction. One way to tackle this issue is to penalise the disparity between each group's average output and the overall average and redefine the objective function as

$$\begin{aligned}\mathscr{L}^{\mathscr{O}}(X,Y) := \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \ell_{i,j}(x_i \cdot y_j, A_{i,j}) \\ + \gamma T\left(u_1(X,Y) - \overline{u}(X,Y), \cdots, u_K(X,Y) - \overline{u}(X,Y)\right) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j),\end{aligned}$$

where $u_k(X,Y) := \frac{\sum_{(i,j)\in\Omega_k} x_i \cdot y_j}{|\Omega_k|}$ is the average outcome in group k and $\overline{u}(X,Y) := \frac{\sum_{(i,j)\in\Omega} x_i \cdot y_j}{|\Omega|}$ is the (possibly reweighed) sample average. In the case of wLSE, thanks to its shift property, this can be rewritten as

$$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left(\ell_{i,j}(x_i \cdot y_j, A_{i,j}) - \gamma(x_i \cdot y_j)\right) + \gamma T\left(u_1(X,Y), \cdots, u_K(X,Y)\right) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j), \tag{5.13}$$

Importantly, this notion preserves the factorisation property of fGLRMS in the sense that non-penalty terms only depend on the dot product $x_i \cdot y_j$ and is still an fGLRM.

5.6.2 Integrating balanced notions of fairness

Additional notions of fairness (Chhabra et al., 2021), specific to clustering, can be investigated and tackled efficiently within the fGLRM framework. In the below, a clustering problem is considered, whereby one wishes to cluster data points from K groups into C clusters.

5.6.2.1 Balance fairness (Chierichetti et al., 2017)

The (reformulated) notion of balance can be implemented in a slightly generalised version of our framework. Indeed, the following objective encourages proportions of points in cluster l to be similar across groups:

$$\mathscr{L}^{\mathscr{B}}(X,Y) := \mathscr{L}(X,Y) + \gamma \sum_{l=1}^{C} T\left(\frac{\sum_{(i,j)\in\Omega_1} \mathbf{1}_{\{x_i=e_l\}}}{|\Omega_1|}, \cdots, \frac{\sum_{(i,j)\in\Omega_K} \mathbf{1}_{\{x_i=e_l\}}}{|\Omega_K|}\right). \tag{5.14}$$

5.6.2.2 Bounded representation (Ahmadian et al., 2019)

Similarly, one can encode a notion such as bounded representation by introducing a new penalty term

$$\mathscr{L}^{\mathscr{R}}(X,Y) := \mathscr{L}(X,Y) + \gamma \sum_{k=1}^{K} \sum_{l=1}^{C} r_{k,l}\left(\frac{\sum_{(i,j)\in\Omega_k} \mathbf{1}_{\{x_i=e_l\}}}{|\Omega_k|}\right), \tag{5.15}$$

where $r_{k,l}(p)$ is worth 0 if $b \leq p \leq a$ and $+\infty$ otherwise. Note that one needs to be careful with the initialisation of an algorithm with bounded representation and may wish to perform stratified sampling per group and per cluster and/or use a smooth representation of $r_{k,l}(p)$.

5.7 Experimental assessment

In this section, the following results are demonstrated.

- **Multiple GLRMs.** PCA, k-means and non-negative matrix factorisation (NMF) are considered in our experiments.

- **Reproducibility and convergence.** Using the proposed wLSE functional with a large positive α (10^5), it is possible to reproduce results from (Ghadiri et al., 2021; Samadi et al., 2018) (which can also be recovered by simply picking $T = \max$ with the alternating minimisation approach).

- **Partial debiasing.** By varying the α hyperparameter in the wLSE functional, a full spectrum of results ranging from the standard GLRM, to intermediate states and the min-max solution is obtained. This points to the usual fairness-accuracy trade-off as the overall average cost tends to increase as disparity decreases.

- **Generalisation.** For each level of α, the corresponding solution calibrated on the train set (i.e., the y_js are kept fixed) is used, and the new set of weights x_i is computed for all is in the test set. This is similar to online dictionary learning. First, the performance on the test set (expectedly) deteriorates and, second, the completely fair solution may have become sub-optimal. This reinforces the attractiveness of partial debiasing, which can thus be interpreted as a fair regularisation.

Note that we consider group disparity as a key metric to monitor in this Section.

5.7.1 Data, models and approaches presented

5.7.1.1 Datasets

Three datasets (whose details are indicated in Table 5.1) were considered:

- German Credit (Dua & Graff, 2017),

- Loan Default Credit (Yeh & hui Lien, 2009),

- Adult (Dua & Graff, 2017).

Table 5.1: The details of binary and multivariate protected attributes in each dataset.

Dataset	Reference	Binary	Multivariate
Adult	(Dua & Graff, 2017)	*sex*; female (16,192), male (32,650)	*race*; (white (41,762), black (4,685), other (406) asian-pac-islander (1,519), amer-indian-eskimo (470)
German Credit	(Dua & Graff, 2017)	*sex*; female (310), male (610)	*sex & marital status*; male : divorced/separated (50), female : divorced/separated/married (392), male : single (548); male : married/widowed (92)
Loan Defaults	(Yeh & hui Lien, 2009)	*sex*; female (11,888), male (18,112)	-
LFW	(Huang et al., 2008)	*sex*; female (2,962), male (10,270)	-

Throughout these experiments, groups have been defined in terms of membership to a class defined thanks to a protected characteristic, as detailed in Table 5.1.

Remark 18. Results in Figures 5.1- 5.2 are based on the aforementioned datasets and in Table 5.1 additional details on the protected attribute used are provided, as well as unique values and their counts. In Appendix A.4.3, some additional results based on the LFW (Labeled Faces in the Wild) dataset (Huang et al., 2008) are indicated.

5.7.1.2 fGLRMs under consideration

In this paper, the following objectives were implemented in the fGLRM framework:

- k-Means (Figures 5.1-5.4),
- Principal Component Analysis (PCA) (Figures 5.5-5.6),
- Non-Negative Matrix factorization (NMF) (Figures A.1-A.2).

It has to be noted that the framework is flexible enough and allows for different modifications.

5.7.1.3 Fairness functionals and benchmarks

In addition to the wLSE functional, a number of benchmark functionals were also leveraged:

- Standard GLRM (i.e., empirical average loss as in (Udell et al., 2016));
- Reweighed GLRM (i.e., uniform weight $1/K$ for each group, similar to (Kamiran & Calders, 2012)), see Figure A.2 in particular;
- Min-max approach (where $T = \max$) as in (Ghadiri et al., 2021; Samadi et al., 2018);
- p-norm (or q-FFL) approach (in line with (Li et al., 2020b), see Figure A.1);

It has been shown that the proposed framework can handle cases with two (see Figures 5.1-5.2) and more protected groups (see Figures 7.8-5.4). Figures 5.5 - 5.6 demonstrate results when using group functional on outcome disparity presented in Section 5.6.1. For each aforementioned result, the trade-off curve that is obtained when varying the α parameter of wLSE functional[1] is indicated.

Finally, in Sections A.4.2 and A.4.3, results based on supervised GLRM and outcome-based fairness incorporating a penalty term discussed in Appendix A.1 are presented.

[1] When q-FFL functional used q parameter is varied instead.

Remark 19. In experiments where a test set is needed, a 70%-30% train-test split was used. Stratified sampling with respect to the protected attribute was used to ensure that both train and test sets have the same proportion of observations belonging to different groups. The algorithm used throughout these tests is the standard alternating minimisation. When considering a grid of values for α, the following values: $10^{-6}, 10^{-5}, 3\times10^{-3}, 10^{-2}, 10^{-1}, 5\times10^{-1}, 1, 5, 10^{1}, 2\times10^{1}, 6\times10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}$ were considered

5.7.2 Main observations

Before delving into the precise results, a summary of the key empirical findings is proposed.

First, considering a standard GLRM or a (partially) debiased one makes a difference both in terms of average statistical performance and disparity, thus indicating the presence of accuracy-fairness trade-offs in unsupervised learning too.

Second, “interpolating” techniques such as wLSE or q-FFL tend to offer similar results and converge to the min-max programme as their respective hyperparameter goes to ∞. In- and out-of-sample behaviour indicates that it is not always preferable to use the min-max formulation.

Third, using a reweighing scheme seems to help improve fairness in general (but not always), especially when α is small. However, as α increases, it becomes less relevant.

Fourth, the implementation within the fGLRM framework of the min-max approach or the adaptation of the q-FFL logic to unsupervised learning are straightforward and match results obtained via other algorithms. This underlines the interest in having a generic framework that accommodates easily many different functionals.

5.7.3 Varying the hyperparameter α

Having established that the fGLRM approach can replicate previous results, we demonstrate some of its further benefits, such as its ability to interpolate between standard (i.e., no fairness considerations) and min-max solutions, with the degree of interpolation being controlled by a parameter α as shown in Equation A.1. Figure 5.1 shows how both total loss and group disparity change on the train set, as we vary parameter α through a grid of values. The larger values of α decrease group disparity, while smaller values bring the solution closer to the standard solution.

In Figures 5.1-5.6, the following notations are used: the "average" point is the solution of a standard algorithm, the "min-max" point corresponds to the solution of wLSE with the largest $\alpha = 10^5$, while all other points are denoted as $wLSE_\alpha$ and correspond to the remaining values of α. The size of dots is ordered according to the α used and the dotted line is a local regression line through the points.

Key observations. Results presented in Figure 5.1 are intuitive: decreasing the disparity amongst groups increases the overall loss, and vice versa, which illustrates the fairness-accuracy trade-off (Kleinberg et al., 2017) in the case of unsupervised learning.

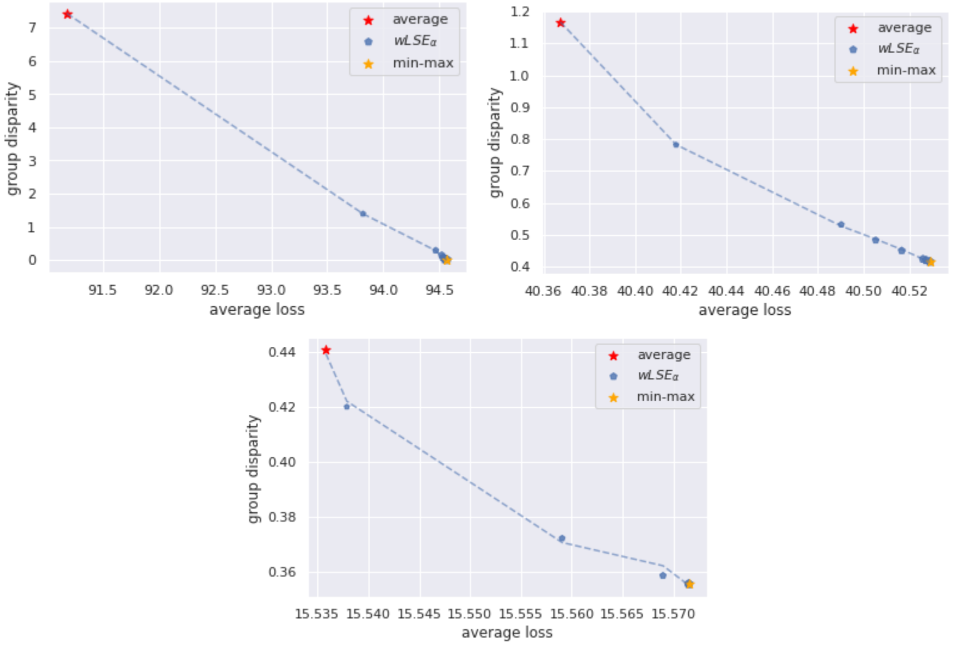

Figure 5.1: KMeans. Trade-off curve between the average loss and group disparity on the *train set* with $wLSE_\alpha$ functional on adult, German credit, and loan defaults' datasets. Each point corresponds to a different α.

5.7.4 Generalisation

To check whether debiasing generalises in fGLRMs, their out-of-sample behaviour was considered and the performance of the fitted fGLRMs was assessed on a test set. The idea here is to keep "archetypes" Y learned on the train set fixed and solve for the best feature representations X_{test} of test set examples: $\frac{1}{|\Omega|}\sum_{j:(i,j)\in\Omega}\ell_{i,j}(x_i\cdot y_j^{train},A_{i,j}^{test})+\sum_{i=1}^{m}r_i(x_i)$. Once these are learned, computing the average loss or group disparity is straightforward. For the sake of brevity, only the k-means algorithm is considered, so that test observations are allocated to the nearest centroid obtained during training. Despite a clear pattern on the train set, it is not always the case on the test set for different data sets as shown in Figure 5.2, in line with results pertaining to supervised learning (Agrawal et al., 2020). Indeed, a “U”-shape is discernible in some test set results (rather than the train set's “L” shape), whereby certain choices of hyper-parameter α result in a higher loss and a higher disparity (i.e., a lose-lose situation). This implies that full debiasing upon training may not be always desirable.

Key observations. This suggests carefully choosing α and using cross-validation techniques, depending on the exact use case.

Remark 20. Additional results are presented in the Appendix A.4.

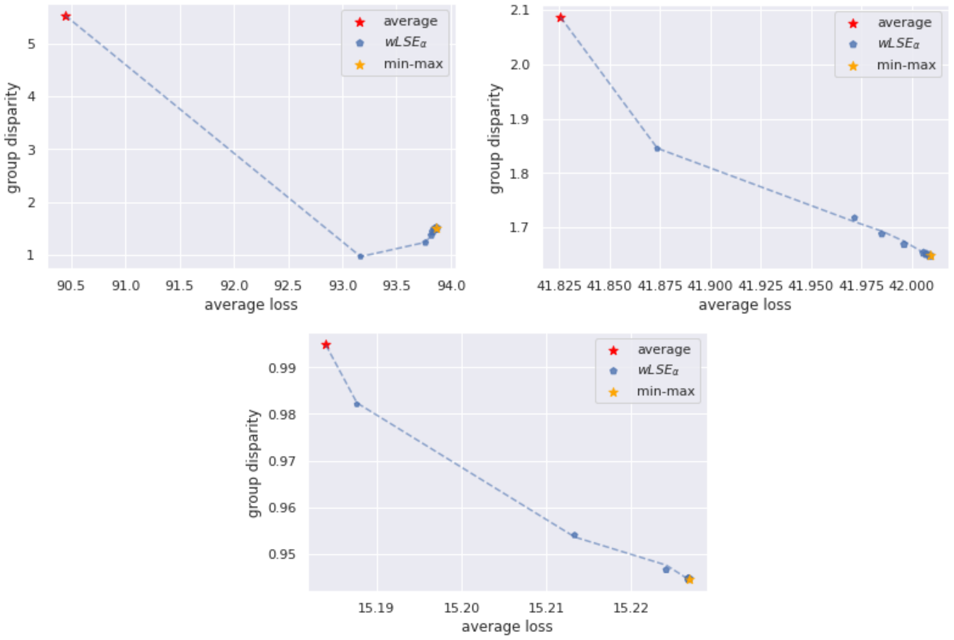

Figure 5.2: KMeans. Trade-off curve between average loss and group disparity on the *test set* with wLSE functional on adult, german credit, and loan defaults' datasets.

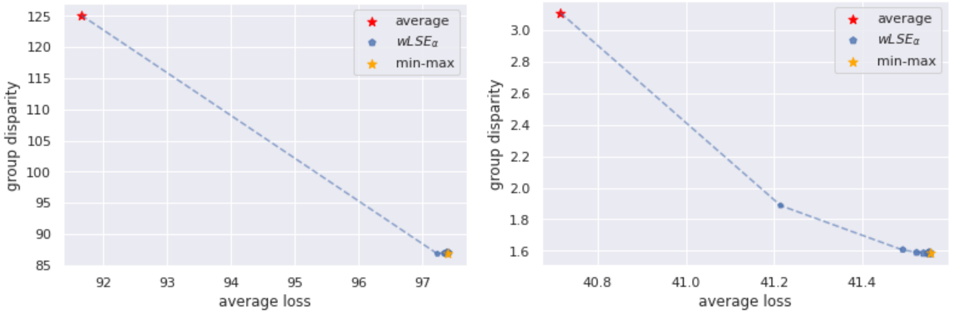

Figure 5.3: KMeans. Trade-off curve between average loss and group disparity on the *train set* with wLSE functional on the Adult and German Credit datasets. The protected attribute is a multivariate feature (race in the Adult dataset, sex and marital status in the German Credit dataset respectively).

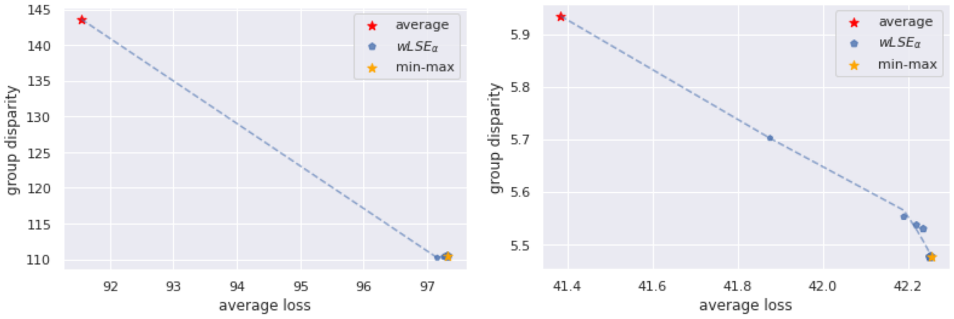

Figure 5.4: KMeans. Trade-off curve between the average loss and group disparity on the *test set* with wLSE functional on the Adult and German Credit datasets. The protected attribute is a multivariate feature (race in the Adult dataset, sex and marital status in the German Credit dataset respectively).

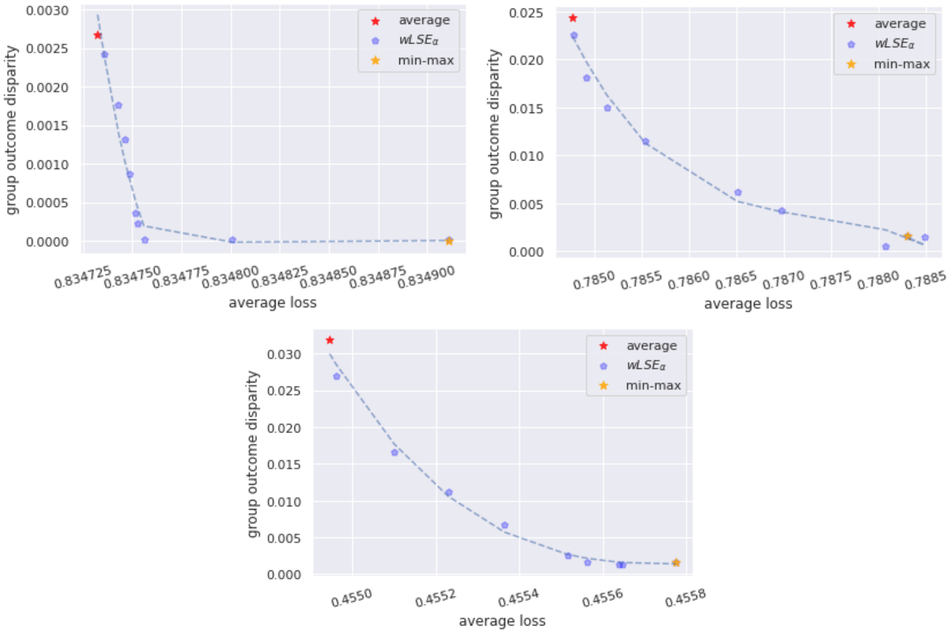

Figure 5.5: PCA. Trade-off curve between the average loss and group outcome disparity on the *train set* with the wLSE functional on the Adult, German Credit, and Loan Defaults datasets.

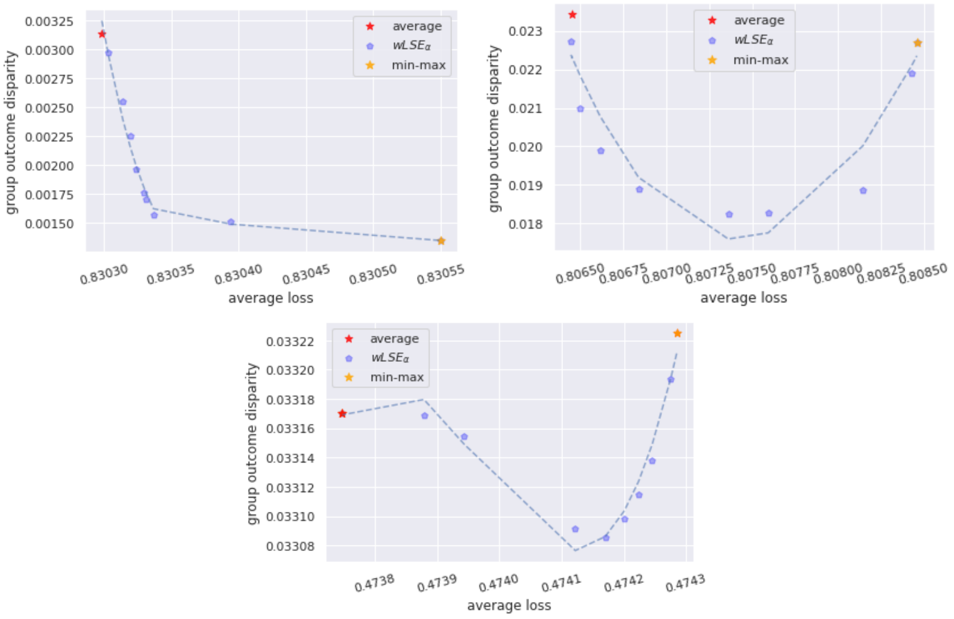

Figure 5.6: PCA. Trade-off curve between the average loss and group outcome disparity on the *test set* with the wLSE functional on the Adult, German Credit, and Loan Defaults datasets.

5.8 Discussion

In this chapter, the notion of fair generalised low-rank models was introduced by applying a fairness functional to group-wise average losses, leading to a reduction in cost disparity across groups. Building fair GLRMs has enabled a generic framework, encompassing fair PCA and fair k-means, that is also applicable to many other use cases, such as sparse PCA, non-negative matrix factorisation, subspace clustering and many more.

A particular choice of such fairness functional was also specified, namely the weighted Log-Sum Exponential, which has many desirable properties. This permits users to select a hyper-parameter α that governs the fairness-accuracy trade-off. The importance of debiasing is emphasised by some of the out-of-sample results presented, showing that total debiasing in-sample may lead to very different results out-of-sample. In addition, it was shown that straightforward algorithms based on biconvexity (or variants thereof) could be efficiently leveraged to solve these fair objective functions. fGLRMs thus inherit some properties of GLRMs.

Finally, some extensions are straightforward, such as including orthogonality constraints between the learnt dictionary and the specified protected characteristic. However, further research is warranted to understand how to transpose multiple fairness definitions from classification or regression to unsupervised learning and how to assess out-of-sample performance.

Part III

Multi-Objective Reinforcement Learning

Chapter 6

Robust Multi-Objective Reinforcement Learning with Dynamic Preferences

This chapter results from a paper co-authored with Parth Pahwa. F.B.G. conceived of the presented ideas, developed the theoretical aspects, designed the experiments, and wrote the manuscript. P.P. contributed to the code base, ran the numerical experiments and co-wrote parts of the underlying paper. Both authors discussed the results and commented on the manuscript.

6.1 Research objectives

This chapter considers multi-objective reinforcement learning (MORL) when preferences over multiple tasks are not perfectly known. Indeed, it is often the case in practice that an agent is trying to achieve tasks that may have competing goals but does not exactly know how to trade them off. The goal of MORL is thus to learn optimal policies under a set of possible preferences leading to different trade-offs on the Pareto frontier. Here, a new method is proposed by considering the *dynamics* of preferences over tasks. While this is a more realistic setup in many scenarios, more importantly, it helps devise a simple and straightforward approach by considering a surrogate state space of both states and preferences, leading to a joint exploration of states and preferences. Static (and possibly unknown) preferences can also be understood as a limiting case of this framework. In sum, this allows devising both deep Q-learning and actor-critic methods based on *planning* under a preference-dependent policy and *learning* the multi-dimensional value function under said policy. Finally, the performance and effectiveness of this method are demonstrated in experiments run on different domains.

6.2 Introduction

Multi-objective reinforcement learning (MORL) is a sub-field of reinforcement learning that deals with decision-making problems with multiple, often conflicting objectives. In traditional reinforcement learning, the goal is to maximise a single scalar reward signal. However, in many real-world problems, such as resource allocation, portfolio optimisation, or robotic control, multiple objectives must be considered simultaneously, such as maximising profit while minimising risk or achieving multiple tasks simultaneously. MORL algorithms attempt to find policies that can balance multiple objectives by optimising a set of trade-offs between them. In MORL, the reward signal is typically a vector of multiple objectives rather than a single scalar value. The agent tries to learn a policy that can maximise some goals while minimising others or achieving a good trade-off. Several approaches to solving MORL problems include Pareto-based, weighted-sum, and goal-attainment methods. In short, the goal of an agent is to learn an optimal policy under a set of preferences expressing the relative importance of each objective. To deal with multi-objective problems, researchers have tried to incorporate preferences as a fixed choice and scalarise rewards (Konak et al., 2006; Lin, 2005; Mossalam et al., 2016), thereby reducing the problem to a single objective optimisation task. While this approach is well-researched, it only addresses the subset of problems where the preferences over the objectives are known beforehand. A tangential strategy is to learn a set of optimal policies that can span the space of preferences. These methods were addressed in (Barrett & Narayanan, 2008; Li et al., 2020a; Natarajan & Tadepalli, 2005), but generally lack scalability in high dimensional environments.

This chapter presents a multi-objective Markov decision process ("MOMDP") framework, incorporating a transition function over preferences, leading to an algorithm capable of learning a single parameterised policy encompassing the dynamics of preferences over tasks. This algorithm can be extended to existing state-of-the-art methods, such as deep Q-learning (Mnih et al., 2013) and actor-critic methods (Mnih et al., 2016), while overcoming the shortcomings of existing MORL methods. It is scalable as it optimises for a single parameterised preference-dependent policy but implements the principles of multi-task learning ("MTL") to learn a multi-dimensional value function under the said policy. Furthermore, since the proposed approach considers a surrogate state space of states and preferences, a *joint* exploration strategy of states and preferences has been devised. Thus, a traditional Q-learning algorithm is supplemented with hindsight experience replay (Andrychowicz et al., 2017) for better sample efficiency and an exemplar network (Fu et al., 2017) for efficient exploration over the surrogate state space. In Section 6.3, background concepts are introduced, and our revised MOMDP with preference transition is described. In Section 6.4, a novel Robust Multi-Objective Reinforcement Learning with Dynamic Preferences ("RDP MORL") algorithm is put forward, along with some theoretical guarantees. Section 6.5 shows some empirical results and benchmarks against other state-of-the-art MORL algorithms.

6.3 Background and related work

6.3.1 Multi-objective Markov decision process

First, the Markov framework for solving the multi-objective sequential decision problem is introduced. An MOMDP can be represented by the tuple $\langle \mathscr{S}, \mathscr{A}, P_S, \mathbf{r}, \gamma, \Omega, f_\omega, P_\omega \rangle$, where:

- $\mathscr{S}$ is the state space,

- $\mathscr{A}$ is the action space,
- $P_S : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to [0,1]$ is the transition function over the state space,
- $\mathbf{r} : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to \mathbb{R}^d$ is the vector reward function specifying rewards for $d \geq 1$ objectives,
- γ is the discount factor and $\gamma \in [0,1)$,
- Ω is the space of preferences where $\omega \in \Omega$ s.t. $\sum_{i=1}^d \omega_i = 1$ and $\omega_i \geq 0$ for $d \geq 1$ objectives,
- $f_\omega : \mathbb{R}^d \to \mathbb{R}$ is the scalarisation function for preference $\omega \in \Omega$,
- $P_\omega : \Omega \times \Omega \to [0,1]$ is the transition function over the preference space.

Note that if $d = 1$ and $P_\omega(\omega, \hat{\omega}) = 0$ if $\omega \neq \hat{\omega}$ and 1 if $\omega = \hat{\omega}$, the MOMDP collapses to a standard Markov decision process ("MDP")[1]. A policy is defined as a mapping $\pi : \mathscr{S} \times \mathscr{A} \to [0,1]$ from state to action. The vector value of state s at time t under a policy π is given by the multidimensional value function, defined as

$$\mathbf{V}^\pi(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbf{r}_{t+i+1} | \pi, S_t = s\right], \tag{6.1}$$

where $\mathbf{r}_{t+1}$ is the reward received at time-step $t+1$. Similarly, the vectorised $\mathbf{Q}$ function can be defined as the expected long-term reward by taking action a in state s at time t under a policy π:

$$\mathbf{Q}^\pi(s,a) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbf{r}_{t+i+1} | \pi, S_t = s, A_t = a\right]. \tag{6.2}$$

If one considers the set of all possible value functions, the Pareto frontier can be constructed $\mathscr{F}^* := \{\mathbf{V}(s) | \nexists \mathbf{V}^{'}(s) \geq \mathbf{V}(s))$ and under the space of all possible (linear) preferences in Ω, the convergence set ("CCS") of the Pareto frontier defined as $\mathscr{CCS} := \{\mathbf{V}(s) \in \mathscr{F}^* | \exists \omega \in \Omega \; s.t. \; \omega^T \mathbf{V}(s) \geq \omega^T \mathbf{V}^{'}(s), \forall \, \mathbf{V}^{'}(s) \in \mathscr{F}^*\}$, where $\omega \in \mathbb{R}^d$ s.t. $\sum_{i=1}^d \omega_i = 1$ and $\omega_i \geq 0$.

Remark 21. The convex convergence set is a subset of the Pareto frontier, i.e., $\mathscr{CCS} \subset \mathscr{F}^*$ and the Pareto front can be regarded as a set of non-dominated policies, i.e., there exists no other policy that can improve the expected return for an objective without reducing the expected return of at least one different objective. Throughout this chapter, for simplicity's sake, only the scenario where the scalarisation function f_ω is linear is presented, i.e. $f_\omega(\mathbf{r}) = \omega^T \mathbf{r}$.

Problem statement For any MOMDP, there exists a set of policies corresponding to the convex convergence set ($\mathscr{CCS}$). Our goal is to train an agent to discover and act according to the preference-dependent policy, which spans the $\mathscr{CCS}$.

6.3.2 Related work

Multi-Objective reinforcement learning MORL problems involve finding Pareto optimal solutions using multi-objective optimisation and reinforcement learning techniques. MORL algorithms follow either a single-policy or a multiple-policy (Vamplew et al., 2011) approach. Single-policy approaches

[1] In the case where we have $d > 1$ objectives and $P_\omega(\omega, \hat{\omega}) = 0$ if $\omega \neq \hat{\omega}$ and 1 if $\omega = \hat{\omega}$, one recovers the classic formulation of the MOMDP.

seek to find the optimal policy by fixed preference-induced scalarisation of the multi-objective problem. Researchers have explored the effects of both linear and non-linear scalarisation (Van Moffaert et al., 2013). However, in reality, the set of preferences may be unknown at training time or may change over time. Multiple-policy approaches focus on approximating the set of policies that span the Pareto frontier. The methodologies range from repeatedly calling single-policy MORL over different preferences (Mossalam et al., 2016; Natarajan & Tadepalli, 2005; Zuluaga et al., 2016), generalising the Q-learning update rule to multi-objective settings (Reymond & Nowé, 2019; Yang et al., 2019) or by modifying gradient based policy search (Parisi et al., 2014; Pirotta et al., 2015a; Pirotta et al., 2015b). Recent works have demonstrated the application of MORL with deep reinforcement learning (Friedman & Fontaine, 2018; van Seijen et al., 2017), in high dimensional state space (Abdolmaleki et al., 2020). In the proposed approach, evaluation is performed on both discrete and continuous observation spaces.

Meta-learning and multi-task learning These methods were explored by (Chen et al., 2019b; Riedmiller et al., 2018; Teh et al., 2017; Wulfmeier et al., 2019) as a way of solving multi-objective control problems. Meta-learning paradigm involves learning a general policy that is not Pareto optimal but computationally efficient and adapting to objective preferences, while multi-task learning frameworks solve the MORL problems by jointly learning a separate policy. While the methodology in this chapter implements multi-task learning, learning tasks are created over the vectorised and scalar-value functions and not the competing objectives.

Q-learning The Q-Learning framework has also been extended to the MORL framework (Mossalam et al., 2016; Yang et al., 2019). In particular, Scalarised Q-learning (Mossalam et al., 2016) uses a vector-value function with scalar updates and searches over preferences. The scalar updates, which involve computing the inner product of the value with the preferences, lead to sample inefficiency and sub-optimal MORL policies. While Envelope Q-learning (Yang et al., 2019) tries to address these shortcomings, our approach introduces the transition function over preferences and learning over the extended state space, leading to robust and faster learning. The key contributions that distinguish the present work, robust Q-learning with dynamic preferences (RDP Q-learning), from (Yang et al., 2019) are (1) the introduction of a transition function over preferences, which allows the agent to adapt to dynamic preferences while it is performing actions in the environment, and (2) the formulation of estimating both the vector and scalar value functions as a multi-task learning problem with shared parameters. Additionally, introducing surrogate state spaces allows efficient exploration of the preference space (generally leading to a faster computation of the value function).

Exploration and exemplar networks Exploration plays a fundamental role in RL systems, and in the original deep Q-learning chapter (Mnih et al., 2013), the authors use ε-greedy exploration to overcome the challenge of sparse reward signals. However, ε-greedy or other undirected exploration methods can be exponential in the depth of the state space ("Efficient exploration in reinforcement learning", 1992). Given that our methodology increases the dimensionality by introducing a surrogate state space, undirected methods do not scale and lead to sub-optimal performance. Additionally, the fundamental idea of multi-objective reinforcement learning is to understand the effect of preferences on agent policies. Therefore, an efficient exploration over the *surrogate* space allows us to explore over the preference space and is sample efficient. Hence, in this new implementation, Q-learning is supplemented with exemplar networks from (Fu et al., 2017) for a sophisticated exploration of the preference space. An ablation study is provided in Appendix B.1 to showcase the performance boost.

6.4 Multi-objective RL with dynamic preferences

A new framework for multi-objective reinforcement learning called Robust Multi-Objective Reinforcement Learning with Dynamic Preferences ("RDP MORL") is thus proposed. The key idea is to consider the dynamics of preferences over tasks and learn over a surrogate state space, defined as a combination of states and preferences. The aim is two-fold: (1) dynamics over preferences introduces robustness by considering disturbances in preferences, (2) creating the surrogate state space allows the agent to explore and approximate the Pareto frontier efficiently. While this framework can be applied to deep Q-learning (Mnih et al., 2013) and actor-critic methods (Mnih et al., 2016), the focus in this chapter lies on the former and robust Q-learning with dynamic preferences is therefore introduced.

Robust Q-learning with dynamic preferences The proposed algorithm uses the vectorised Q-value function to allow both the (vector and scalar) Q-networks to learn a set of policies simultaneously. While (Yang et al., 2019) uses a similar methodology, the convex envelope update and optimality filter they define render their approach high-dimensional and computationally complex. This chapter's proposed RDP Q-learning overcomes this hurdle by taking actions according to the scalarised Q network during both learning and evaluation. This methodology results in an algorithm that outperforms competitor algorithms with fewer iterations.

6.4.1 Revisiting MORL

In addition to the previous discussion, two more considerations led to the proposed algorithm. Firstly, preferences can shift over time (Guiso et al., 2018). Secondly, current MORL algorithms do not exploit *directly* the fact that similar preferences should lead to similar Q-values (except at points of discontinuity).

6.4.1.1 Building a surrogate state space

The key insight is to consider a surrogate (or augmented) state space combining the state space $\mathscr{S}$ as well as the preference space Ω

$$\langle \mathscr{S}, \mathscr{A}, P_S, \mathbf{r}, \gamma, \Omega, f_\omega, P_\omega \rangle = \langle (\mathscr{S} \times \Omega), \mathscr{A}, P_S \otimes P_\omega, f_\omega(\mathbf{r}), \gamma \rangle \tag{6.3}$$

In particular, under the linear preference assumption, the reward function in the surrogate space can be defined as $r((s,\omega),a) = \mathbf{r}(s,a)^T \omega$. This leads directly to a straightforward definition of the action-value function Q as

$$Q^{\pi}((s,\omega),a) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^t \mathbf{r}_{t+i+1}^T \omega_{t+i+1} \middle| \pi, S_t = s, \omega_t = \omega, A_t = a\right]. \tag{6.4}$$

Remark 22. Importantly, in general, $Q^{\pi}((s,\omega),a) \neq \mathbf{Q}^{\pi}((s,\omega),a)^T \omega$ as preferences are themselves dynamic. In the particular case where $P(\omega_{j+1} = \omega' | \omega_j = \omega) = \delta(\omega, \omega')$, the equality is recovered.

6.4.1.2 Implicit Bellman operator

Having defined the surrogate state space $(\mathscr{S} \times \Omega)$ and the Q-value function, one can apply Q-learning to the surrogate MDP. In other words, it follows directly from standard Q-learning (Sutton & Barto, 2018) that the multi-objective optimality operator $\mathscr{T}$ can be defined as:

$$\begin{aligned}(\mathscr{T}Q)\,((s,\omega),a) &:= \mathbf{r}(s,a)^T \omega + \gamma \mathbb{E}_{(s',\omega') \sim P_S \otimes P_\omega(\cdot|s,\omega,a)}\left[\max_{a' \in \mathscr{A}} Q((s',\omega'),a')\right] && (6.5)\\ (\mathscr{T}Q)\,(\hat{s},a) &= \mathbf{r}(\hat{s},a) + \gamma \mathbb{E}_{\hat{s}' \sim P_S \otimes P_\omega(\cdot|\hat{s},a)}\left[\max_{a' \in \mathscr{A}} Q(\hat{s},a')\right]. && (6.6)\end{aligned}$$

This paves the way for the use of standard Q-learning, as the Q-value update at iteration $j+1$ can be written as:

$$Q_{j+1}(\hat{s}_j,a) \leftarrow (1-\alpha_j(\hat{s}_j,a))Q_j(\hat{s}_j,a) + \alpha_j(\hat{s}_j,a)\left[\mathbf{r_j}^T \omega_j + \gamma \max_{a'} Q_j(\hat{s}_{j+1},a')\right], \tag{6.7}$$

where $\alpha_j \in [0,1)$ is the chosen step size function.

Remark 23. While the present focus lies on Q-learning, similar approaches to SARSA or policy gradients (and actor-critic methods) are possible.

One of the conditions for a Q-learning algorithm to converge (Melo, 2001) is that $\sum_j \alpha_j(\hat{s},a) = +\infty$ and $\sum_j \alpha_j(\hat{s},a)^2 < +\infty$, for all $\hat{s}, a$. In other words, all state-preference-action triplets must be visited infinitely often. This thus represents a challenge in effectively exploring the space of preferences.

6.4.1.3 Multi-objective Q-value function

The optimal policy derived from Q-learning is $\pi^*(a|\hat{s}) = \mathbf{1}_{\{\arg\max_{a \in A} Q(\hat{s},a)\}}$. To obtain the multivariate Q-value function, all modellers have to do is apply policy evaluation under π^*:

$$\mathbf{Q}_{j+1}(\hat{s}_j,a) \leftarrow (1-\alpha_j(\hat{s}_j,a))\mathbf{Q}_j(\hat{s}_j,a) + \alpha_j(\hat{s}_j,a)\left[\mathbf{r_j} + \gamma \mathbf{Q}_j(\hat{s}_{j+1}, \arg\max_{a' \in A} Q_j(\hat{s},a'))\right]. \tag{6.8}$$

Importantly, these updates can be carried out in parallel with Q-learning on the surrogate state space.

6.4.2 Learning algorithm

Two Q networks are implemented: a multi-objective Q network and a policy Q network. The multi-objective Q-network aims to approximate the vectorised $\mathbf{Q}$-function to allow the network to learn simultaneously over multiple preferences while the agent acts according to the policy selection Q-network. Since both networks observe the dynamics of the same environment, both learning stages are modelled as separate but still related tasks. Multi-task learning is known to outperform single-task algorithms (Zhang & Yang, 2021) in these scenarios. Using the multi-task approach, the algorithm for RDP Q-learning is provided in Algorithm 3.

6.4.2.1 Multi-task Q-learning

Let $\mathscr{L}^{\alpha}(\theta)$ be the loss associated with the Multi-Objective Q-network parameterised by θ and let $\mathscr{L}^{\beta}(\hat{\theta})$ be the loss associated with the policy Q-network parameterised by $\hat{\theta}$ where $\theta, \hat{\theta} \in \Theta$ ($\hat{\theta}$ is a sub-vector of θ). The multi-objective Q-network loss is defined as:

$$\mathscr{L}^{\alpha}(\theta) = \mathbb{E}_{\hat{s},a}\left[||\mathbf{y} - \mathbf{Q}(\hat{s}, a; \theta)||_2^2\right], \tag{6.9}$$

where $\hat{s} = (s, \omega)$ is the surrogate state, $\mathbf{y} = \mathbb{E}_{\hat{s}'}[\mathbf{r} + \gamma \mathbf{Q}(\hat{s}', \hat{a}; \theta)], \hat{a} = \max_a Q(\hat{s}_{j+1}, a; \hat{\theta})$, which is estimated by sampling transitions from the replay buffer. The policy Q-network loss is defined as:

$$\mathscr{L}^{\beta}(\hat{\theta}) = \mathbb{E}_{\hat{s},a}\left[(y - Q(\hat{s}, a; \hat{\theta}))^2\right] \tag{6.10}$$

where $\hat{s} = (s, \omega)$ is the surrogate state and $y = \mathbb{E}_{\hat{s}'}[\omega^T \mathbf{r} + \gamma \max_{\hat{a}} Q(\hat{s}', \hat{a}; \hat{\theta})]$. This leads to a multi-task loss function for the overall network. A framework for solving this is developed in the following sections.

Algorithm 3 RDP Q-learning

Require: network $Q_{\hat{\theta}}$ and $\mathbf{Q}_{\theta}$, sampling distribution G_{ω}, transition distribution H_{ω}, replay buffer D.

```
for $episode = 1, ....N$ do
  Sample a linear preference $\omega_{\mathbf{0}} \sim G_{\omega}$
  for $t = 0, \cdots, M-1$ do
    Observe state $s_t$ and construct surrogate state space $\hat{s}_t = (s_t, \omega_{\mathbf{t}})$
    $a_t = \begin{cases} \text{random action,} & \text{w.p. } \varepsilon \\ \text{arg max}_a Q(\hat{s}_t, a; \hat{\theta}), & \text{w.p. 1-}\varepsilon \end{cases}$
    Execute action $a_t$ and observe reward $\mathbf{r_t}$, state $s_{t+1}$ and sample $\omega_{t+1} \sim H_{\omega_t}$ [a]
    Store transition $(s_t, \omega_t, \mathbf{r_t}, s_{t+1}, \omega_{t+1})$
    Sample random minibatch of transitions of size $N_{\tau}$ $(s_t, \omega_t, \mathbf{r_t}, s_{t+1}, \omega_{t+1})$
    for $i = 1, \cdots, N_{\omega}$ do
      $\omega_i \sim G_{\omega}$
      $\omega_{i+1} \sim H_{\omega_i}$
      for $j = 1, \cdots, N_{\tau}$ do
        $\hat{s}_{i,j+1} = (s_{j+1}, \omega_{i+1})$
        $y_{ij} = \begin{cases} \mathbf{r}_j^T \omega_i, & \text{for terminal s}_{j+1} \\ \mathbf{r}_j^T \omega_i + \gamma \max_{\hat{a}} Q(\hat{s}_{j+1}, \hat{a}; \hat{\theta}), & \text{otherwise} \end{cases}$
        $\hat{\mathbf{y}}_{ij} = \begin{cases} \mathbf{r}_j, & \text{for terminal s}_{j+1} \\ \mathbf{r}_j + \gamma \mathbf{Q}(\hat{s}_{j+1}, \hat{a}; \theta), \hat{a} = \max_a Q(\hat{s}_{j+1}, a; \hat{\theta}) & \text{otherwise} \end{cases}$
      end for
    end for
    Update $Q_{\hat{\theta}}$ and $\mathbf{Q}_{\theta}$ by performing gradient descent according to equation 6.12.
  end for
end for
```

[a] We simulate H_{ω_t} using a Dirichlet distribution with $\alpha = \omega$

6.4.2.2 An adversarial setup

An adversarial approach to multi-task learning is proposed to motivate the current methodology. Let K be the number of tasks, z_k represent the loss associated with task $k = 1, \cdots, K$ and w_k the corresponding weight.

Robustness to uncertainty Given a vector of average individual task losses $z \in \mathbb{R}^K$ and a reference distribution over tasks $w \in \mathbb{S}^{K-1}$, the adversary maximises the overall loss but is constrained by the distance (chosen to be the Kullback-Leibler divergence here) to the reference distribution w. The adversary's problem can thus be written as

$$\max_{\delta \in \mathbb{S}^{K-1}} \delta \cdot z - \frac{1}{\eta} D_{\mathrm{KL}}(\delta || w), \tag{6.11}$$

where $\eta > 0$ is fixed. It is immediate to check that the solution vector is given by $\delta_k^* = \frac{w_k e^{\eta z_k}}{\sum_{k=1}^K w_k e^{\eta z_k}}$ for $k = 1, \cdots K$. Thus, the adversary's problem is recast as a robust optimisation problem and expressed in terms of distribution uncertainty (Glasserman & Xu, 2014).

Model fitting Considering $\mathscr{L}^\alpha(\theta)$, the loss associated with the multi-objective Q-network parameterised by θ, and $\mathscr{L}^\beta(\hat{\theta})$ the loss associated with the policy Q-network parameterised by $\hat{\theta}$, where $\theta, \hat{\theta} \in \Theta$ ($\hat{\theta}$ is a sub-vector of θ), the learning problem is framed in an adversarial setting, leading to the following formulation, where $\eta > 0$, $w_k \geq 0$ for all k and $\sum_{k=1}^K w_k = 1$:

$$\begin{aligned} \min_{\theta, \hat{\theta}} \max_{\delta} \left(\delta_1 \mathscr{L}^\alpha(\theta) + \delta_2 \mathscr{L}^\beta(\hat{\theta}) - \frac{1}{\eta} \left[\delta_1 \log\left(\frac{\delta_1}{w_1}\right) + \delta_2 \log\left(\frac{\delta_2}{w_2}\right) \right] \right) \\ = \min_{\theta, \hat{\theta}} \frac{1}{\eta} \log\left(w_1 e^{\eta \mathscr{L}^\alpha(\theta)} + w_2 e^{\eta \mathscr{L}^\beta(\hat{\theta})} \right). \end{aligned} \tag{6.12}$$

6.4.3 Model architecture

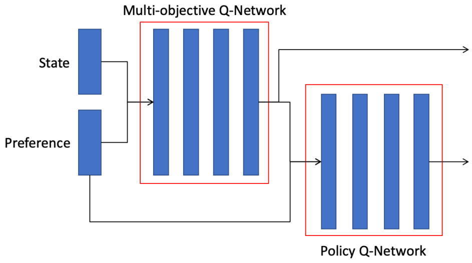

Figure 6.1: Network Architecture

This algorithm uses two different networks, unlike (Mossalam et al., 2016) and (Yang et al., 2019). Since both networks interact with the same environment and are interdependent, a shared network structure is introduced. The shared network creates embeddings that benefit from the joint learning experience. It consists of four fully connected hidden layers with $20 \times (dim(\mathscr{S}) + d)$ hidden units each, where $\mathscr{S}$ is the state vector and d is the number of objectives. The input to the shared network is the surrogate state (i.e., the concatenation of the state and preference vector). The multi-objective Q-network stacks one output layer of size $d \times |\mathscr{A}|$ on top of the shared network where $|\mathscr{A}|$ is the cardinality of the action space. The policy Q-network takes as input the output of the multi-objective Q-network combined with the preference vector and contains four fully connected hidden layers with $20 \times d \times (|\mathscr{A}| \times +1)$ and the output is of size $|\mathscr{A}|$, see Figure 6.1.

6.5 Experiments

In this Section, we evaluate the performance of RDP Q-learning on three multi-objective reinforcement learning problems. It shows how the algorithm can recover the optimal solution in the CCS before comparing its performance against relevant benchmarks.

Evaluation metrics Two metrics are used to evaluate the proposed approach's empirical performance on the problem domains:

- **Coverage ratio** ("CR") evaluates the agent's ability to recover the solutions from the finite convex convergence set. Let m be the number of objectives and $\mathscr{S} \subseteq \mathbb{R}^m$ the set of vector value functions recovered by the algorithm. We define $\mathscr{S} \cap_\varepsilon CCS = \{\mathbf{V}^\pi \in \mathscr{S} | \exists \mathbf{V}^{\pi^*} \in CCS \text{ s.t. } ||\mathbf{V}^\pi - \mathbf{V}^{\pi^*}||_1 / ||\mathbf{V}^{\pi^*}||_1 \leq \varepsilon\}$ Then the Coverage Ratio is calculated as the F-score, where $Precision = |\mathscr{S} \cap_\varepsilon CCS| / |\mathscr{S}|$ and $Recall = |\mathscr{S} \cap_\varepsilon CCS| / |CCS|$.

$$CR(S) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6.13}$$

- **Expected utility metric** ("EUM") evaluates the agent's ability to maximise user utility. It is defined as the expected maximum utility under the solution set S approximated by the algorithm:

$$EUM = \mathbb{E}[\max_{\pi_\omega \in S} f_\omega(\mathbf{V}^{\pi_\omega})] \tag{6.14}$$

Baselines RDP Q-learning's performance is also benchmarked against its peer MORL algorithms: (1) *Envelope Q-learning* (Yang et al., 2019), a state-of-the-art MORL algorithm that uses envelope Q updates to learn multiple policies simultaneously. It modifies deep Q-Network for vector $\mathbf{Q}$ values. (2) *Scalarised Q-learning* (Mossalam et al., 2016), which uses scalarised Q-updates. (3) *MOFQI* (Castelletti et al., 2012), i.e., a multi-objective fitted Q-iteration with a large linear model as Q-approximator. (4) *CN+OLS* (Abels et al., 2019), which is a conditional neural network using an optimistic linear support method.

6.5.1 Fruit tree navigation

The Fruit tree navigation ("FTN") environment is a full binary tree of depth d ($d = 5, 6$ or 7), with a randomly assigned vector reward $\mathbf{r} \in \mathbb{R}^6$ on the leaf nodes, which are the terminal state. The reward vector encodes the values of six nutrition components {Protein, Carbs, Fats, Vitamins, Minerals, Water} in the leaf nodes. The rewards are designed to be Pareto optimal such that, for every leaf node, ω for which its reward is optimal; therefore, all leaves lie on the CCS. The objective associated with the environment is to find a path from the root to a leaf node that maximises our overall utility for a given preference. At any non-terminating state in the tree, the agent has two actions available, choosing between the *left* or *right* subtree.

Table 6.1: Fruit Tree Coverage Ratio ($depth = 5$)

N_ω	Scalarised Q-learning	Envelope Q-learning	RDP Q-learning (2500 episodes)
1	0.9363 ±0.023	0.9706 ±0.027	**0.9980 ± 0.005**
4	0.9840 ±0.016	**1.0000 ± 0.000**	**1.0000 ± 0.000**
8	0.9968 ±0.007	**1.0000 ± 0.000**	**1.0000 ± 0.000**
16	**1.0000 ± 0.000**	**1.0000 ± 0.000**	**1.0000 ± 0.000**

Table 6.2: Fruit Tree Coverage Ratio ($depth = 6$)

N_ω	Scalarised Q-learning	Envelope Q-learning	RDP Q-learning (3000 episodes)
1	0.6250 ±0.057	0.9240 ±0.051	**0.9500 ± 0.012**
4	0.7654 ±0.077	0.9856 ±0.004	**0.9908 ± 0.008**
8	0.8560 ±0.067	0.9808 ±0.007	**0.9912 ± 0.009**
16	0.8976 ±0.062	0.9952 ±0.021	**0.9984 ± 0.003**

The proposed model's performance is benchmarked against Envelope Q-learning and Scalarised Q-learning. All three algorithms are trained on the FTN environment ($depth = 5, 6$ and 7) for 5000 episodes and N_ω sampled preferences during the learning process. The coverage ratio is calculated by testing the performance over 2000 episodes with randomly sampled preferences. The mean results over five trials for different depths are indicated in Tables 6.1, 6.2, 6.4. One can see that RDP Q-learning is sample efficient and can outperform the baselines, even when trained on significantly fewer episodes: 2500 for depth 5, 3000 for depth 6 and 3000 for depth 7.

6.5.2 Deep sea treasure

Deep sea treasure (“DST"), a classic MORL benchmark, is an episodic problem which was created to highlight the limitations of scalarisation (Vamplew et al., 2011). The environment is a 10×11 treasure hunt grid, with the agent controlling a submarine. There are multiple treasure locations with variable treasure values and two associated objectives, (1) minimise the time taken to reach the treasure and (2) maximise the value of the treasure. The treasure values used in this exercise were provided in (Yang et al., 2019) to ensure that the Pareto frontier is convex. For each episode, the agent is placed on the top left corner of the grid and has four available actions: *Up, Down, Left, Right*. The reward received by the agent is a two-element vector, where the first element is the *time penalty* (computed by adding -1 on all turns), and the second element is the *treasure value*. The episode terminates when the agent reaches a treasure state.

All agents are trained for 2000 episodes, and we evaluate each algorithm for 2000 episodes with randomly sampled preferences. The mean coverage ratio results over 5 trials are provided in Table 6.4. The RDP Q-learning has the best coverage ratio and is able to achieve after training for 1850 episodes. Figure 6.2 visualizes the optimal convex convergence set and RDP Q-learning approximated CCS. Notice that the RDP Q-learning solutions cover the entire optimal CCS.

Table 6.3: Fruit Tree Coverage Ratio ($depth = 7$)

N_ω	Scalarised Q-learning	Envelope Q-learning	RDP Q-learning (3000 episodes)
1	0.5847 ±0.061	0.6000 ±0.029	**0.8728 ± 0.021**
4	0.6969 ±0.057	0.6544 ±0.066	**0.8034 ± 0.020**
8	0.6837 ±0.097	0.7437 ±0.040	**0.8326 ± 0.020**
16	0.6532 ±0.029	0.7936 ±0.015	**0.8243 ± 0.017**

Table 6.4: Deep Sea Treasure

Method	Reference	Coverage Ratio
Envelope Q-learning	(Yang et al., 2019)	0.994 ±0.001
Scalarised Q-learning	(Mossalam et al., 2016)	0.989 ±0.024
CN+OLS	(Abels et al., 2019)	0.751 ±0.163
MOFQI	(Castelletti et al., 2012)	0.639 ±0.421
RDP Q-learning		**1.000 ± 0.000**

6.5.3 Mountain car

The multi-objective version of the classic mountain-car task (Sutton, 1995) was first introduced in (Vamplew et al., 2011). In the classical setting, the agent aims to escape the car from the valley in a minimum number of steps. The agent can perform three different actions: (1) not accelerate, (2) accelerate to the right and (3) reverse to the left. Since the car's engine is less powerful than gravity, the agent must reverse to the left to build enough potential energy to escape from the right end. In the single objective setting, the agent receives a reward of -1 for all non-terminating states.

(Vamplew et al., 2011) extends the objective space by introducing two additional objectives: minimise the number of (1) forward and (2) reverse accelerations and introduce a three-dimensional vector reward where a penalty of -1 is received whenever one of the acceleration actions is executed. To increase the complexity of the optimisation problem, an updated reward structure is introduced to provide positive reinforcement when the car displaces in the direction of the action the agent performs.

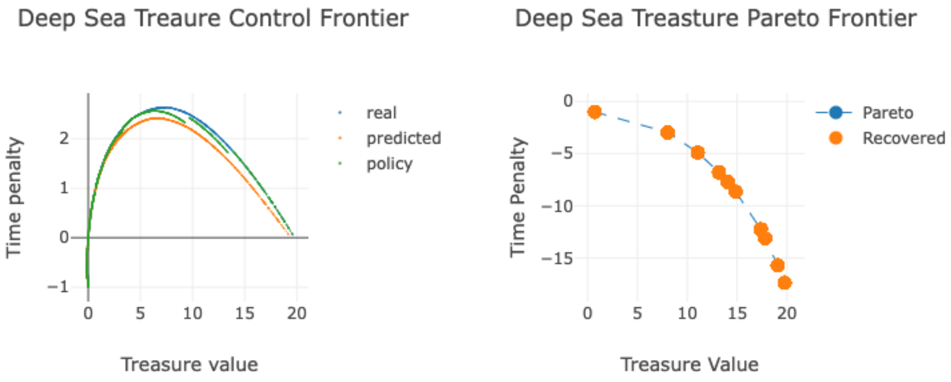

Figure 6.2: Illustration of the true and the RDP Q-learning recovered CCS for the deep sea treasure environment.

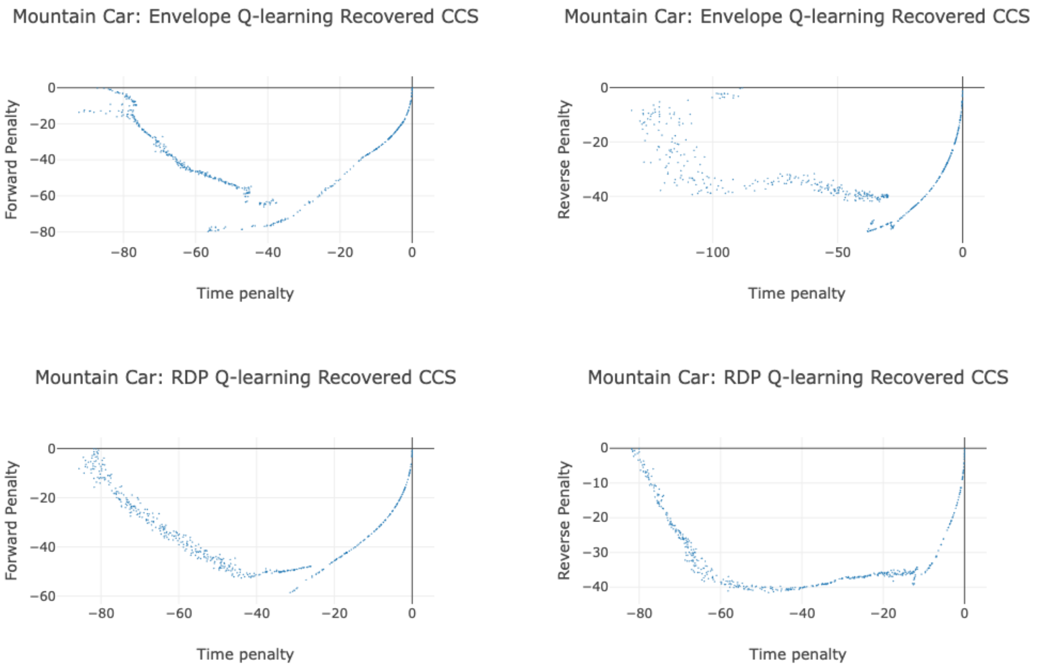

Figure 6.3: Illustration of the Envelope Q-learning and RDP Q-learning recovered CCS for the Mountain Car environment created by randomly sampling 500 preferences and calculating mean value over 50 trials

Table 6.5: Mountain Car: Reward Structure

Time Penalty	Reverse Penalty	Acceleration Penalty	Action	Displacement
-1	0	0	No acceleration	None
-1	0	0	No acceleration	Left
-1	0	0	No acceleration	Right
-1	0	-1	Left	None
-0.5	0.5	-0.5	Left	Left
-1	0	-1	No acceleration	Right
-1	-1	0	Right	None
-0.5	-0.5	0.5	Right	Right
-1	-1	0	Right	Left

The reward structure is defined in Table 6.5. Unlike the FTN and DST environments, the state space of Mountain Car is continuous and has an unknown Pareto frontier, hence performance is assessed empirically by using the expected utility metric.

Both algorithms are trained for 1500 episodes and evaluated on different sets of preferences. The mean results over one hundred trials are provided in Tables 6.5-6.6. One can see that across different preferences RDP Q-learning has a higher expected utility value. The action behaviour is fairly consistent in the case of RDP Q-learning, whereas Envelope Q-network deviates from the optimal behaviour for the preference vector $[0.5, 0.5, 0.0]$ (time, forward acceleration, reverse), by performing a higher count of forward acceleration (right action). The recovered CCS across the 3 objective pairs for both the RDP Q-learning and the baseline is depicted in Figure 6.3. The recovered CCS for the

Table 6.6: Mountain Car: Expected Utility for different preferences

RDP Q-learning		Envelope Q-learning		Preference		
EUM	Steps	EUM	Steps	Time	Forward	Reverse
-77.42	105.72	-81.18	114.30	0.9	0.05	0.05
-28.58	300.00	-31.02	300.00	0.1	0.8	0.1
-27.53	300.00	-30.64	300.00	0.1	0.1	0.8
-27.27	169.79	**-11.57**	235.62	0.0	0.5	0.5
-66.48	125.88	-73.72	132.70	0.5	0.5	0.0
-56.91	169.79	-57.57	148.39	0.5	0.0	0.5

Table 6.7: Mountain Car: Count of actions performed for different preferences

RDP Q-learning			Envelope Q-learning			Preference		
Left	Right	None	Left	Right	None	Time	Forward	Reverse
40.41	65.31	0.00	37.16	77.02	0.12	0.9	0.05	0.05
254.43	0.00	47.57	298.07	0.00	3.93	0.1	0.8	0.1
2.41	201.84	97.75	0.00	296.15	5.85	0.1	0.1	0.8
32.03	45.29	92.47	14.33	17.82	203.47	0.0	0.5	0.5
66.65	59.11	0.12	53.10	71.89	7.71	0.5	0.5	0.0
20.63	112.80	3.50	20.41	125.80	2.18	0.5	0.0	0.5

baseline is neither smooth nor convex, whereas RDP Q-learning retrieves a more consistent CCS.

6.6 Conclusion

In this chapter, a multi-objective reinforcement learning (MORL) framework was proposed, which considers a surrogate (or augmented) state space comprising both states and preferences over the objectives. By designing an implicit Markov Decision Process based on this surrogate state space, *dynamic* preferences are made possible. In addition, learning and planning via Q-learning under this particular formulation then become possible. As noticed in previous research, exploring the space of preferences is crucial in deriving optimal policies (and approximating the Pareto frontier), which is achieved here by encouraging the *joint* exploration of states and preferences. This is further facilitated by adding exemplar rewards. This approach turns out to be particularly sample-efficient and robust. Finally, the effectiveness of the proposed framework was demonstrated by achieving improved performance against other state-of-the-art MORL algorithms in three different environments. Further research points to extending this approach to other algorithms, such as the actor-critic methods and exploring the impact of various types of preference dynamics on policy choices.

Chapter 7

Multi-Objective Multi-Armed Bandits via Gaussian Process Upper Confidence Bounds

This chapter results from two papers co-authored with Peter Hill. F.B.G. conceived of the presented ideas, developed the theoretical aspects, designed the experiments, and wrote the manuscript. P.H. contributed to the code base, ran the numerical experiments and co-wrote parts of the underlying papers. Both authors discussed the results and commented on the manuscript.

Research objectives

The consideration of different rewards simultaneously is often desired in a multi-armed bandit setting. For instance, when optimising marketing campaigns, more than one key metric may be of interest to make informed decisions. This leads to the concept of multi-objective multi-armed bandits. This chapter introduces a new framework for efficiently solving multi-objective multi-armed bandit problems using Gaussian Processes. Gaussian Processes provide a flexible non-parametric approach to model the overall reward function and have thus been extensively employed in multi-armed bandit settings. The framework presented in this paper extends prior work on the Gaussian Process Upper Confidence Bound algorithm to a multi-objective setting, taking into account preferences over objectives. Moreover, the proposed approach handles varying levels of observability, including cases where different amounts of reward information are known. This framework allows effective learning from all available information.

7.1 Introduction

Multi-armed bandits (Robbins, 1952) are a class of machine learning problems in which an agent must decide which of several actions to take to maximise its cumulative reward over time. The term "multi-armed bandit" comes from the analogy to a slot machine or "one-armed bandit," where a gambler must decide which of several slot machines to play in order to maximise their winnings. In a multi-armed bandit problem, the agent is presented with a set of actions or "arms," each of which has an associated reward distribution. The agent must decide which arm to pull at each time step, based on the observed rewards from previous pulls, with the goal of maximising its total reward over a fixed time horizon. The key challenge in multi-armed bandit problems is the trade-off between exploration and exploitation. On the one hand, the agent needs to explore different arms to learn their reward distributions and identify the arm with the highest expected reward. On the other hand, the agent needs to exploit the best arm it has identified to maximise its cumulative reward. There are many algorithms for solving multi-armed bandit problems (Kuleshov & Precup, 2014), including epsilon-greedy, Upper Confidence Bound (UCB), and Thompson Sampling. These algorithms balance exploration and exploitation in different ways, and their performance depends on the properties of the reward distributions and the specific problem setting.

However, there are also cases where we may have multiple reward objectives that we wish to consider, for example, in a campaign optimisation setting, where a user may be interested in increasing impressions and clicks in some ratio. This leads to a *multi-objective multi-armed bandit problem*, where these multiple objectives need to be traded off amongst themselves, according to the user's preferences. To accomplish this, the idea of a surrogate reward is introduced, allowing for the scalarisation of the rewards' vector across different objectives.

To solve these problems, the use of *Gaussian Processes* (GP's) (Rasmussen, 2003) to model the reward function is put forward. They allow modellers to flexibly model the multi-objective reward function in a non-linear, non-parametric way. GP's are very useful when modelling with limited data and noisy samples. They also allow dealing with a high-dimensional input space.

7.1.1 Background on multi-armed bandits

Previous research on multi-objective multi-armed bandits generally falls into the category of scalarisation, where a function transforms the environment into a single-objective environment to which standard single-objective approaches can be applied. (Drugan & Nowe, 2013) proposed multiple extensions to the "UCB1" algorithm (Auer et al., 2002), to include multi-objectives via scalarisation and Pareto search[1].

Significant research has also been carried out into using Gaussian processes in multi-armed bandit problems, focused on the single-objective setting, which is leveraged as a building block in this chapter. The Gaussian Process Upper Confidence Bound algorithm (GP-UCB) (Srinivas et al., 2009) was the first algorithm of its kind, providing an approach for modelling the reward function using a Gaussian Process. (Krause & Ong, 2011) furthered this algorithm by introducing contextual information, and (Bogunovic et al., 2016) and (Imamura et al., 2020) created time-varying variants, which provide the ability to forget data as it gets older and therefore less relevant.

Combining these two topics, (Dai et al., 2020) look at applying GP-UCB to a multi-objective setting by optimising for each objective separately, using multi-output Gaussian Processes. (Swersky et al., 2013) solve this problem simultaneously across objectives, using an entropy search strategy

[1] UCB stands for Upper Confidence Bound.

to maximise the information gain over the location of the minimum of the reward function. This chapter's approach builds on the contextual bandits' algorithms by considering a multi-objective setting with user preferences through task functionals and exploring different levels of observability.

In addition, (Yahyaa & Manderick, 2015) extended Thompson Sampling (Thompson, 1933) for use in multi-objective multi-armed bandit problems, whilst (Vakili et al., 2021) propose an approach based on Sparse Gaussian Processes, to help with the scalability of the Thompson Sampling approach.

7.1.2 Contributions

In this chapter, building upon the work previously referenced, a generic framework for solving multi-objective multi-armed bandits is delineated using Gaussian Processes. The proposed approach has some key benefits over existing algorithms:

- The incorporation of the user's preferences towards the objectives and contextual information can be achieved. This allows actions to be chosen that trade off between objectives based on these preferences. The preferences can either be a fixed input by the user or explored by our algorithm over time.

- The idea of *task functionals* is employed to enable the generalisation of the relationship between the user's preference and the rewards and the concept of a *surrogate reward* is introduced, which encapsulates the space of rewards.

- The concept of *observability* is considered, and in a multi-objective setting, the observation data received may change, unlike in a single-objective setting where a single reward value for a given action is always received. The proposed algorithm can adapt to different levels of reward observation data and learn from all the information available.

The proposed framework converges with sub-linear regret, showing this algorithm's improvement compared with other approaches through experimentation.

7.2 Problem statement

7.2.1 Multivariate rewards

Suppose one wishes to optimise an unknown multi-objective reward function $f : \mathscr{X} \to \mathbb{R}^J$, where J objectives are considered. Here f is a function of the action $\mathbf{s} \in \mathscr{S}$ and any contextual information $\mathbf{z} \in \mathscr{Z}$. In (Krause & Ong, 2011), they consider $\mathscr{X} := \mathscr{S} \times \mathscr{Z}$ to be the input action–context space. This could be a constrained space. Here, $f(\mathbf{s}_t, \mathbf{z}_t) = (f^{[1]}(\mathbf{s}_t, \mathbf{z}_t), \cdots, f^{[J]}(\mathbf{s}_t, \mathbf{z}_t))^T \in \mathbb{R}^J$ is defined to be the reward vector. In a multi-objective setting, the idea of a reward differs from that of a single-objective setting, where there is always only a scalar reward function. In the multi-objective case, a separate reward could be defined for each objective, or a single reward could be defined, combining the different objectives.

7.2.2 Game-theoretic setup

In a multi-armed bandit problem, one iteratively samples from our reward function and each time observe a noisy reward value $\mathbf{y}_t$ as follows:

$$\mathbf{y}_t = f(\mathbf{s}_t, \mathbf{z}_t) + \varepsilon_t, \tag{7.1}$$

where $\mathbf{y}_t = (y_t^{[1]}, \cdots, y_t^{[J]})^T \in \mathbb{R}^J$, and $\varepsilon_t = (\varepsilon_t^{[1]}, \cdots, \varepsilon_t^{[J]})^T \in \mathbb{R}^J$. The $\varepsilon_t^{[j]}$'s are independent and identically distributed Gaussian noise $\varepsilon_t^{[j]} \sim N(0, \sigma^2)$. The aim is to learn the optimal action to take to maximise the reward.

7.2.3 Preference space

Since the problem is multi-objective, the user may consider certain objectives more important than others. The user preference space, $\mathscr{U}$, is defined as the space in which the user considers the importance of different objectives. For example, one could consider $\mathscr{U} = \mathbb{S}_{J-1} := \{\mathbf{a} \in \mathbb{R}^J : a_j \geq 0, \sum_j a_j = 1\}$. Thus, we define our space also to include preferences, that is, $\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$. Crucially, this allows learning the reward function and optimising it under different chosen trade-offs (note that this also covers the case of dynamic preferences).

7.2.3.1 Task Functionals

Here, the concept of a task functional is introduced, which can heuristically be understood as a mapping between preferences and the related optimisation objective.

Definition 12. Let $\mathscr{F} = \mathscr{F}^{[1]} \times \cdots \times \mathscr{F}^{[J]}$ be the space of true rewards, where $\mathscr{F}^{[i]}$ is the space for the i-th reward. One can define a *multi-task functional*,

$$\mathscr{T} : \mathscr{U} \times \mathscr{F} \to \mathbb{R}, \tag{7.2}$$

to be a function of $\mathbf{u} \in \mathscr{U}$ and $f \in \mathscr{F}$.

7.2.3.2 Surrogate reward function

Equipped with this definition, a surrogate reward function can now be constructed that takes preferences $\mathbf{u}$, as well as actions $\mathbf{s}$ and context $\mathbf{z}$, as inputs.

Definition 13. The *surrogate reward function* $\tilde{f}$ is defined as

$$\tilde{f} : \mathscr{X} \to \mathbb{R}, \tag{7.3}$$

which is a preference-dependent single-value reward function such that

$$\tilde{f}(\mathbf{s}, \mathbf{u}, \mathbf{z}) = \mathscr{T}(\mathbf{u}, (f^{[1]}(\mathbf{s}, \mathbf{z}), \cdots, f^{[J]}(\mathbf{s}, \mathbf{z}))), \tag{7.4}$$

where $\mathscr{T}$ is a task functional.

Importantly, this leads to defining the noisy observed reward as

$$\tilde{y}_t = \tilde{f}(\mathbf{s}_t, \mathbf{u}_t, \mathbf{z}_t) + \varepsilon_t, \tag{7.5}$$

for $\varepsilon_t \in \mathbb{R}$. Preferences can change through time or be randomly selected, chosen by the user, or selected by a given logic (such as maximal exploration of the preference space).

Examples The following are examples of a preference space and task functional that could be used:

- *Scalarisation*: $\mathscr{U} = \mathbb{S}_{J-1}$ and $\mathscr{T}(\mathbf{u}, f) = \sum_{j=1}^{J} u_j f^{[j]}(\mathbf{s}, \mathbf{z})$. [2]
- *Combined*: $\mathscr{U} = \{\mathbf{e_k} : k = 1, \cdots 3\}$ [3] and $\mathscr{T}(\mathbf{u}, f) = [\min_j f^{[j]}, \max_j f^{[j]}, \sum_j f^{[j]}]^T \mathbf{u}$
- *Gini Index* (Busa-Fekete et al., 2017): $\mathscr{U} = \mathbb{S}_{J-1}$ and $\mathscr{T}(\mathbf{u}, f) = -G_u(\mathbf{s})$, where $G_u(\mathbf{s}) = \mathbf{u}^T \mathbf{s}_\sigma$, and $\mathbf{s}_\sigma$ is obtained by sorting $\mathbf{s}$ into non-decreasing order, and u is non decreasing.

The case $|\mathscr{U}| = 1$ covers the scenario in which preferences are fixed and cannot change, i.e., scalarisation[4].

7.2.4 The Role of observability

A particular aspect of multi-objective settings is the presence of additional information. Indeed, it is usually the case, for instance, that the result of choosing $\mathbf{s}_t$ at step t can be observed for all objectives $j = 1, \cdots, J$, and not only the surrogate noisy, preference dependent, reward function $\tilde{y}$. Thus, the *information set* available at time t is introduced:

$$\overline{\mathscr{O}}_{0:t} := \bigcup_{0<t' \leq t} \mathscr{O}_{t'}, \tag{7.6}$$

where the time-objective pair subset at time t' is introduced as

$$\mathscr{O}_{t'} := \left\{ y_{t'}^{[j]} \text{ s.t. } j \in \{1, \cdots J\} \text{ and } y_{t'}^{[j]} \text{ is observable} \right\}. \tag{7.7}$$

In short, $\overline{\mathscr{O}}_{0:t}$ represents the additional observable quantities during the game. Note that one could observe differing amounts of additional information each time we sample from the reward function. Then, the overall information available at step t is considered to be $\overline{\mathscr{F}}_{0:t}$, where

$$\overline{\mathscr{F}}_{0:t} = \overline{\mathscr{O}}_{0:t} \cup \{\tilde{y}_{t'} : t' \leq t\}. \tag{7.8}$$

[2] This is the approach taken by (Shen et al., 2022), with $u_j = 1$.

[3] $\mathbf{e_k}$ is the standard basis vector: $\mathbf{e_k} = (0, \cdots, 0, 1, 0, \cdots 0)^T$, with 1 at position k and 0's elsewhere.

[4] The proposed approach also includes existing algorithms in the area of *combinatorial multi-armed-bandits* (Chen et al., 2013), where one chooses k from n arms, $\{s_1, \cdots, s_n\}$, at each iteration. We consider the space $\mathscr{S} = \{(s_{i_i}, \cdots, s_{i_k}) \,|\, i_j \in \{1, \cdots, n\}\}$ to be the $\binom{n}{k}$ combinations of k arms. Suppose $s = \{s_{i_1}, \cdots s_{i_k}\} \in \mathscr{S}$. $u_s \in \mathscr{U}$ is dependent on s and so $\mathscr{U} = \{u \,|\, u = \sum_{j=1}^{k} \mathbf{e}_{s_{i_j}}\}$. Then $\mathscr{T}(\mathbf{u}, f) = \sum_{j=1}^{J} u_j f^{[j]}(\mathbf{s}, \mathbf{z})$.

Note that if $\overline{\mathscr{O}}_{0:t} = \emptyset$, then one returns to the single-objective multi-armed bandit setting. In the following, we assume that the surrogate reward is always observed but that other task-specific rewards may or may not be observed.

7.2.5 Gaussian processes

The use of Gaussian Processes to model the reward function f is put forward in the following. Firstly, a new space is defined for the reward function, allow to consider extra information by considering the specific objective $J+1$ in this space. A larger space $\tilde{\mathscr{X}}$ is defined, where the space of objectives (or tasks) $\mathscr{J}$ is also included:

$$\tilde{\mathscr{X}} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z} \times \mathscr{J}. \tag{7.9}$$

For example, one could consider $\mathscr{J} = \{\mathbf{e_j} : j = 1, \cdots, J+1\}$, or alternatively $\mathscr{J} = [1, \cdots, J+1]$. Thus, $|\mathscr{J}| = J+1$.

Remark 24. The previously defined space $\mathscr{X}$ can be seen as a subspace embedded in $\tilde{\mathscr{X}}$, which can be embedded by setting a task $j = J+1$ in $\tilde{\mathscr{X}}$, i.e.,

$$\mathscr{X} = \left\{\mathbf{x} = (\mathbf{s}, \mathbf{u}, \mathbf{z}, j) \in \tilde{\mathscr{X}} \text{ s.t. } j = J+1\right\}. \tag{7.10}$$

In the case where no additional (task-specific) observations are available, then $\mathscr{X} = \tilde{\mathscr{X}}$.

Crucially, at each time step, the proposed model gets updated with rewards $\mathscr{O}_t$, while an action is chosen based on the surrogate reward, $\tilde{y}$. The space $\mathscr{J}$ gives this algorithm a reference for the rewards for each objective, and thus can use all information whilst still optimising for $\tilde{y}$. Figure 7.1 highlights the proposed approach.

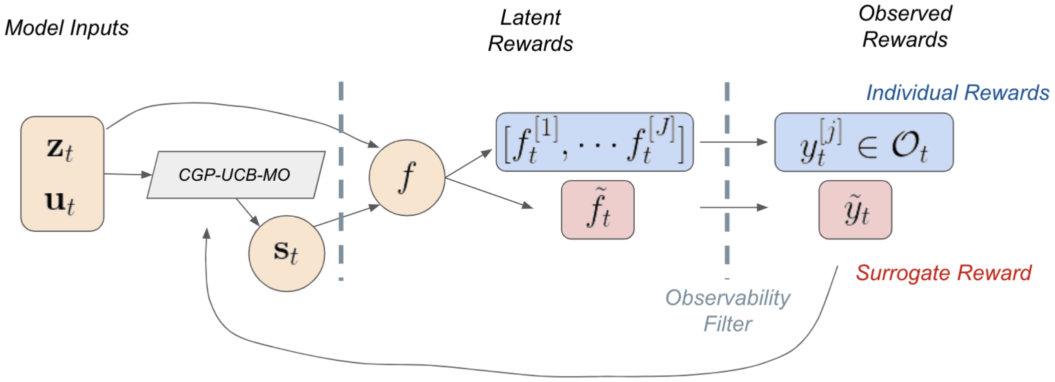

Figure 7.1: Flowchart of the CGP-UCB-MO approach. Note that $\tilde{f}_t$ is the same as $f_t^{[J+1]}$.

The reward function $\tilde{f} : \tilde{\mathscr{X}} \to \mathbb{R}$ is defined to be a function on $\tilde{\mathscr{X}}$. $\tilde{f}$ can be modelled using a Gaussian Process. This is a task-, preference- and context-dependent reward, and thus *all* the information available can be used $\overline{\mathscr{F}}_{0:t}$ at step t in the model. Further, a point in the space $\tilde{\mathscr{X}}$ is denoted as $\mathbf{x} = (\mathbf{s}, \mathbf{u}, \mathbf{z}, \mathbf{j}) \in \tilde{\mathscr{X}}$.

Definition 14. A Gaussian Process, $\mathbf{y}(\mathbf{x})$, is an infinite collection of random variables such that every finite subset, $\mathbf{y}(\mathbf{x}_1), \cdots, \mathbf{y}(\mathbf{x}_T)$, follows a multivariate Gaussian distribution, where $\mathbf{x}_t \in \tilde{\mathscr{X}}$ and $\mathbf{y}(\mathbf{x}_t) \in \mathbb{R}^J$ for all $t = 1, \cdots, T$.

A Gaussian Process can be defined in terms of a mean function $\mu(\mathbf{x})$, and a covariance function $\mathbf{K}(\mathbf{x}, \mathbf{x}')$. Then it comes

$$\begin{pmatrix} \mathbf{y}(\mathbf{x}_1) \\ \vdots \\ \mathbf{y}(\mathbf{x}_T) \end{pmatrix} \sim N \left(\begin{pmatrix} \mu(\mathbf{x}_1) \\ \vdots \\ \mu(\mathbf{x}_T) \end{pmatrix}, \begin{pmatrix} \mathbf{k}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_1, \mathbf{x}_T) \\ \vdots & \ddots & \vdots \\ \mathbf{k}(\mathbf{x}_T, \mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_T, \mathbf{x}_T) \end{pmatrix} \right). \tag{7.11}$$

The interested reader is referred to (Williams & Rasmussen, 2006) for further details. Using Gaussian Processes allows enforcing specific properties on the underlying function, such as smoothness, by definition of the kernel of the GP, whilst maintaining a non-parametric model. Suppose that one has $X = \{\mathbf{x}_1, \cdots \mathbf{x}_T\}$ points with corresponding values $\mathbf{Y}_T = [y_1, \cdots, y_T]^T$. One of the key properties of Gaussian Processes is that the posterior distribution of a new point $y(\mathbf{x})$ is also Gaussian, which allows for a closed form expression for μ and $\mathbf{K}$ as follows:

$$\mu_T(\mathbf{x}) = \mathbf{k}_T(\mathbf{x})^T (\mathbf{K}_T + \sigma^2 I)^{-1} \mathbf{Y}_T \tag{7.12}$$

$$\mathbf{k}_T(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_T(\mathbf{x})^T (\mathbf{K}_T + \sigma^2 I)^{-1} \mathbf{k}_T(\mathbf{x}') \tag{7.13}$$

for $x, x' \in \mathscr{X}$, and where $\mathbf{K}_T$ is a positive definite matrix such that $(\mathbf{K}_T)_{(i,j)} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $k_T(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \cdots, k(\mathbf{x}, \mathbf{x}_T)]^T$.

7.3 Regret

A key quantity in multi-armed bandit algorithms is regret. In a *single objective setting*, one considers the regret r_t at step t, based on input $\mathbf{s}_t$ to be

$$r_t := \sup_{\mathbf{s} \in \mathscr{S}} f(\mathbf{s}, \mathbf{z}_t) - f(\mathbf{s}_t, \mathbf{z}_t) \tag{7.14}$$

for $t = 1, \cdots, T$ and the cumulative regret R_T to be $R_T := \sum_{t=1}^{T} r_t$. Similarly, the average regret is defined as $\overline{R}_T := R_T / T$. These allow measuring the proposed algorithm's learning speed; if R_T grows sublinearly in T, then $\overline{R}_T \to 0$ as $T \to +\infty$.

7.3.1 Multi-objective regret

In a multi-objective setting, the definition of regret is a non-trivial one. One definition could be considering the surrogate reward function $\tilde{f} \in \mathbb{R}$ and defining regret and cumulative regret as above.

In this case, the surrogate regret could be defined as

$$\tilde{r}_t := \sup_{s \in \mathscr{S}} \tilde{f}(\mathbf{s}, \mathbf{u}_t, \mathbf{z}_t) - \tilde{f}(\mathbf{s}_t, \mathbf{u}_t, \mathbf{z}_t), \tag{7.15}$$

with $\tilde{R}_T := \sum_{t=1}^{T} \tilde{r}_t$. However, one may also wish to define regret in terms of any extra information we obtain.

7.3.1.1 Regret decomposition

Consider now the consequences of such a definition on entry-wise regret. First, the regret for task j, $r_t^{[j]}$ is defined as

$$r_t^{[j]} := \sup_{s \in \mathscr{S}} f^{[j]}(\mathbf{s}, \mathbf{z}_t) - f^{[j]}(\mathbf{s}_t, \mathbf{z}_t). \tag{7.16}$$

Proposition 21. *The regret $r_t^{[j]}$ can be decomposed into the following components:*

$$r_t^{[j]} = \left[\sup_{s \in \mathscr{S}} f^{[j]}(\mathbf{s}, \mathbf{z}_t) - f^{[j]}(\mathbf{s}_t^*, \mathbf{z}_t) \right] + \left[f^{[j]}(\mathbf{s}_t^*, \mathbf{z}_t) - f^{[j]}(\mathbf{s}_t, \mathbf{z}_t) \right], \tag{7.17}$$

where $\mathbf{s}_t^* = \arg\max_{\mathbf{s} \in \mathscr{S}} \tilde{f}(\mathbf{s}, \mathbf{u}_t, \mathbf{z}_t)$.

The first term is the regret due to solving a multi-objective problem rather than just task j (it is thus irreducible), whereas the second term represents the regret due to choosing a sub-optimal $\mathbf{s}_t$.

Remark 25. In the case of *linear preferences*, where $\tilde{f}(\mathbf{s}, \mathbf{u}, \mathbf{z}) = \mathbf{u}^T f(\mathbf{s}, \mathbf{z})$, with $\mathbf{u} \in \mathbb{S}_{J-1}$ one can directly analyse the relationship between the regret bounds on f and those on $f^{[j]}$, for $j = 1, \cdots, J$. It is easy to see that $\tilde{r}_t(\mathbf{u}) = \sup_{s \in \mathscr{S}} \tilde{f}(\mathbf{s}, \mathbf{u}, \mathbf{z}_t) - \tilde{f}(\mathbf{s}_t, \mathbf{u}, \mathbf{z}_t) \leq \sum_{j=1}^{J} u_j r_t^{[j]}$. Further, $\sup_{\mathbf{u} \in \mathbb{S}_{J-1}} \tilde{r}_t(\mathbf{u}) = \max_{j=1 \cdots, J} r_t^{[j]}$.

7.4 CGP-UCB-MO

This Section proposes the new CGP-UCB-MO algorithm, followed by a discussion of preferences and kernel choices.

7.4.1 CGP-UCB

The framework leveraged by the proposed approach is the previously mentioned *Contextual Gaussian Process Upper Confidence Bound* algorithm (Krause & Ong, 2011). This is a single-objective algorithm which models the reward function as a Gaussian process and updates the reward function by choosing the input $\mathbf{s}_t$ at time t such that

$$\mathbf{s}_t := \operatorname{argmax}_{\mathbf{s} \in \mathscr{S}} \mu_{t-1}(\mathbf{s}, \mathbf{z}_t) + \sqrt{\beta_t} \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t), \tag{7.18}$$

where $\mathbf{z}_t$ is the contextual information at time t. Here, there is a trade-off between exploitation (the mean term, μ_{t-1}) and exploration (the variance term, σ_{t-1}).

7.4.2 Making CGP-UCB multi-objective

As mentioned, the new reward function $\bar{f}$, which is task-, preference- and context-dependent, is modelled as a Gaussian Process. At each step t, it is assumed that the following is observed:

$$\begin{aligned}
\tilde{y}_t &= \tilde{f}(\mathbf{s}_t, \mathbf{u}_t, \mathbf{z}_t) + \varepsilon_t && (7.19)\\
y_t^{[j]} &= f^{[j]}(\mathbf{s}_t, \mathbf{z}_t) + \varepsilon_t, && (7.20)
\end{aligned}$$

for j such that $y_t^{[j]} \in \mathscr{O}_t$. Here, $f_t^{[j]}$ could be a function of the preference vector $\mathbf{u}_t$, but that is not necessarily the case. Define $\mathbf{x}_t^{[j]} = (\mathbf{s}_t, \mathbf{u}_t, \mathbf{z}_t, j)$ with corresponding observation $y_t^{[j]}$ and use each observation as a separate input when updating our Gaussian Process. Thus, at each iteration t, the model is updated with $1 + |\mathscr{O}_t|$ data points. For example, suppose the reward for each objective is observed. In that case, the data used for the overall update is $\mathbf{x}_t^{[j]} = [\mathbf{s}_t, \mathbf{z}_t, \mathbf{u}_t, j]$ with reward $y_t^{[j]}$ for $j = 1, \cdots J$, and crucially also $\mathbf{x}_t^{[J+1]} = [\mathbf{s}_t, \mathbf{z}_t, \mathbf{u}_t, j_{J+1}]$ with surrogate reward $y_t^{[J+1]} = \tilde{y}_t$.

By modelling $\bar{f}$ from the observed noisy samples, it is possible to learn the true trade-off function $\tilde{f}$ and the individual task rewards themselves. For example, one could pick $\mathbf{u} = \mathbf{e_j} \in \mathscr{U}$, then $\tilde{f}(\mathbf{s}_t, \mathbf{e_j}, \mathbf{z}_t) = f^{[j]}(\mathbf{s}_t, \mathbf{z}_t)$. When multiple observations are available, it is also possible to learn their joint distribution. Algorithm 4 shows the proposed approach for learning when modelling $\bar{f}$ as a Gaussian process and by leveraging the CGP-UCB algorithm. Note, crucially, that an action is chosen with respect to the *surrogate reward*, given by task $J+1$.

```
Algorithm 4 CGP-UCB-MO Algorithm
Require: $\mathbf{z_t}$, input space $\mathscr{S}$
    Set preference vector, $\mathbf{u}_t$.
    Choose $\mathbf{s}_t = \text{argmax}_{\mathbf{s} \in \mathscr{S}} \mu_{t-1}(\mathbf{s}, \mathbf{u}_t, \mathbf{z}_t, J+1) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(\mathbf{s}, \mathbf{u}_t, \mathbf{z}_t, J+1)$
    Observe $\tilde{y}_t$ and $y_t^{[j]} \in \mathscr{O}_t$
    Update $\mu$ and $\sigma$ based on update rules in Equations 7.12 and 7.13.
```

Remark 26. Note here, that if $\mathscr{O}_t = \emptyset$, then the proposed approach boils down to applying the standard CGP-UCB algorithm to $\tilde{f}$.

Remark 27. Suppose that preferences were not included in the space, and instead, an independent Gaussian process was built for each preference. Denote by $\tilde{f}^{[\mathbf{u}]}$ the surrogate reward function when considering preference $\mathbf{u}$. By applying CGP-UCB to $\tilde{f}^{[\mathbf{u}]}$ for each $\mathbf{u}$ individually, where only $\mu_t^{[\mathbf{u}]}$ and $\sigma_t^{[\mathbf{u}]}$ would be updated when considering $\mathbf{u}$, one would obtain the Scalarised Multi-Objective UCB algorithm proposed by (Drugan & Nowe, 2013), except using Gaussian Processes.

Discussion on preferences There is significant flexibility in how modellers can choose to set the preference vector, $\mathbf{u}_t$, in this algorithm. $\mathbf{u}_t$ could be either set by the user at each iteration or selected to explore the space of preferences. The latter could be useful if one wishes to derive all possible trade-offs between preferences and would allow one to learn a Pareto frontier. To achieve this, it is further proposed to either randomly select $\mathbf{u}_t$ at the start of every iteration, or choose using $\mathbf{u}_t = \text{argmax}_{\mathbf{u}} \sigma_{t-1}(\mathbf{s}_{t-1}, \mathbf{u}, \mathbf{z}_{t-1})$, which allows exploring the space more rapidly, and as discussed in (Srinivas et al., 2009), using this update rule maximises the information gain.

7.4.3 Kernel choice

A key part of using Gaussian processes is the choice of kernel. In this case, since the kernel is defined on $\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$ (or on $\tilde{\mathscr{X}} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z} \times \mathscr{J}$ in the presence of additional task-level observations), it is possible, crucially, to learn across different preferences vectors. An avenue of choice is to consider composite kernels (Duvenaud et al., 2011; Williams & Rasmussen, 2006), including additive or multiplicative kernels.

7.4.3.1 Composite kernel and co-regionalisation

A multiplicative kernel of the form:

$$k\left((\mathbf{s}, \mathbf{z}, \mathbf{u}, j), (\mathbf{s}', \mathbf{z}', \mathbf{u}', j')\right) = k_S(\mathbf{s}, \mathbf{s}') k_Z(\mathbf{z}, \mathbf{z}') k_U(\mathbf{u}, \mathbf{u}') k_J(j, j')$$

is proposed. One can also consider a *co-regionalisation kernel*, as proposed by (Bonilla et al., 2007), based on the task j:

$$k((\mathbf{s}, \mathbf{z}, \mathbf{u}, j), (\mathbf{s}', \mathbf{z}', \mathbf{u}', j') = k_S(\mathbf{s}, \mathbf{s}') k_Z(\mathbf{z}, \mathbf{z}') k_U(\mathbf{u}, \mathbf{u}') \mathbf{B}_{j,j'},$$

where $\mathbf{B} \in \mathbb{R}^{J \times J}$ is a positive semi-definite matrix with an entry for each task-pair combination[5].

7.5 Regret analysis

In this Section, the performance of the proposed algorithm is investigated by considering information gains and bounds derived from it.

7.5.1 Information gain

Per (Krause & Ong, 2011; Srinivas et al., 2009), recall the definition of the information gain, γ:

$$\gamma(T; k; \mathscr{X}) := \max_{A \subset S : |A| = T} I(\mathbf{y}_A; f), \tag{7.21}$$

where $I(\mathbf{y}_A; f) = H(\mathbf{y}_A) - H(\mathbf{y}_A | f)$ quantifies the decrease in Shannon entropy (i.e., uncertainty) about f achieved by revealing $\mathbf{y}_A$. Importantly, in the proposed framework, the observations $y_{\mathbf{A}}$ depend on the action-preference-context triplet $\mathbf{x} = (\mathbf{s}, \mathbf{u}, \mathbf{z})$, and $f : \mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z} \to \mathbb{R}$ is the payoff function over the action-preference-context space. It is well-known in the case of a GP that

$$I(\mathbf{y}_A; f) = \frac{1}{2} \log |\mathbf{I} + \sigma^{-2} \mathbf{K}_A|, \tag{7.22}$$

[5] Gaussian processes work well with limited quantities of data. In the very high dimensional case, their performance can reduce. This could be handled using kernel approximation techniques for the inverse of the kernel matrix (See chapter 8 of (Williams & Rasmussen, 2006)), but this is left for further research.

where $\mathbf{K}_A = [k(\mathbf{x},\mathbf{x}')]_{\mathbf{x},\mathbf{x}'\in A}$ is the kernel (or Gram) matrix associated with the kernel k assessed on the set $A \subset S$.

7.5.2 Bounds on information gains from composite kernels

Here, the relationship between the information gain of the composite kernel and that of its components is examined. Recall that regret relative to $\tilde{f}$ is considered, hence on $\mathscr{X}$ only.

Proposition 22. *Let k_S, k_U and k_Z be kernel functions on $\mathscr{S}$, $\mathscr{U}$ and $\mathscr{Z}$ respectively. Then the additive kernel function $k = k_S \oplus k_U \oplus k_Z$ defined on $\mathscr{X}$ verifies*

$$\gamma(T;k_S \oplus k_U \oplus k_Z;\mathscr{X}) \leq \gamma(T;k_S;\mathscr{S}) + \gamma(T;k_U;\mathscr{U}) + \gamma(T;k_Z;\mathscr{Z}) + 4\log T. \tag{7.23}$$

Furthermore, let k_U and k_Z be kernel functions on $\mathscr{U}$ and $\mathscr{Z}$ respectively, with respective ranks at most d_U and d_Z, and such that $k_U(\mathbf{u},\mathbf{u}') \leq 1$ and $k_Z(\mathbf{z},\mathbf{z}') \leq 1$. Then the multiplicative kernel function $k = k_S \otimes k_U \otimes k_Z$ defined on $\mathscr{X}$ satisfies

$$\gamma(T;k_S \otimes k_U \otimes k_Z;\mathscr{X}) \leq d_U d_Z \gamma(T;k_S;\mathscr{S}) + 2d_U d_Z \log T. \tag{7.24}$$

Proof. The results follow from the repeated use of Theorems 2 and 3 in (Krause & Ong, 2011), as well as the associativity of the Kronecker product. □

Intuitively, this suggests that adding preferences increases the information gain.

7.5.3 Observability decreases the information gain

Conditionally on selecting the observation points $\mathbf{x}_t$ for $t = 1,\cdots,T$, the information gain can be expressed in terms of the predictive variances (cf. Lemma 4.2 in (Krause & Ong, 2011)):

$$I(\mathbf{y}_T;f) = \frac{1}{2}\sum_{t=1}^{T} \log\left(1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)\right). \tag{7.25}$$

Proposition 23. *If two kernels, k and $\mathring{k}$, are given, with the same observations points, then the difference in information gain is:*

$$I(\mathbf{y}_T;f;k) - I(\mathbf{y}_T;f;\mathring{k}) = \frac{1}{2}\sum_{t=1}^{T} \log\left(\frac{1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)}{1+\sigma^{-2}\mathring{\sigma}_{t-1}^2(\mathbf{x}_t)}\right). \tag{7.26}$$

In particular, if $\mathring{\sigma}_{t-1}^2(\mathbf{x}_t) \leq \sigma_{t-1}^2(\mathbf{x}_t)$ for all t's, then $I(\mathbf{y}_T;f;\mathring{k}) \leq I(\mathbf{y}_T;f;k)$. Finally, this leads to

$$\gamma(T;\mathring{k};\mathscr{X}) \leq \gamma(T;k;\mathscr{X}). \tag{7.27}$$

Proof. See Appendix. □

This result is key to understanding the role of adding observations as it decreases the information gain. No correlation between the individual rewards gives no added information, whilst a perfect correlation implies that the full observation setting applies. Thus, loosely speaking, observing T iterations and K tasks per iteration yields as much information as observing TK iterations in a single observation setting.

Application to the proposed setting Figure 7.2 shows the decomposition of the kernel matrix, which we define as $\mathring{K}_t$, into different parts, based on observability, at step $t-1$. Firstly, it is important to note that this is a valid kernel matrix. The matrix $K_{t-1}^{(\mathscr{O})}$ corresponds to the kernel entries for the extra information that is observed, through $\overline{\mathscr{O}}_{t-1}$, whilst the matrix $K_{t-1}^{(\tilde{y})}$ corresponds to the entries from the surrogate reward observations. We note that the matrix is block-symmetric. The matrix $K_{t-1}^{(\mathscr{O},\tilde{y})}$ is the kernel entries for cross-observation types, for observations up to step $t-1$. The update rule for the entire kernel as previously discussed reads

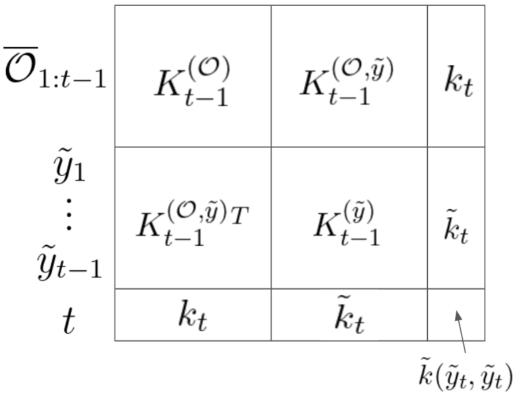
$\overline{\mathcal{O}}_{1:t-1}$
$K_{t-1}^{(\mathcal{O})}$
$K_{t-1}^{(\mathcal{O},\tilde{y})}$
k_t
$\tilde{y}_1$
$\vdots$
$\tilde{y}_{t-1}$
$K_{t-1}^{(\mathcal{O},\tilde{y})T}$
$K_{t-1}^{(\tilde{y})}$
$\tilde{k}_t$
t
k_t
$\tilde{k}_t$
$\tilde{k}(\tilde{y}_t,\tilde{y}_t)$

Figure 7.2: Decomposition of the Kernel Matrix K_t in Equation 7.13.

$$\mathring{\sigma}_{t-1}^2(\mathbf{x}_t) = k(\mathbf{x}_t,\mathbf{x}_t) - \mathring{k}_t^T\left(\mathring{K}_t + \sigma^2 I\right)^{-1}\mathring{k}_t. \tag{7.28}$$

If no extra information were observed, then the kernel matrix would reduce to $K^{(\tilde{y})_{t-1}}$. In the case where only the surrogate task is observed, the update rule is simply

$$\sigma_{t-1}^2(\mathbf{x}_t) = k(\mathbf{x}_t,\mathbf{x}_t) - \tilde{k}_t^T\left(K_t^{(\tilde{y})} + \sigma^2 I\right)^{-1}\tilde{k}_t. \tag{7.29}$$

Now, using the kernel matrix decomposition, define

$$\mathring{K}_t = \begin{pmatrix} K_{t-1}^{\mathscr{O}} & K_{t-1}^{(\mathscr{O},\tilde{y})} \\ K_{t-1}^{(\mathscr{O},\tilde{y})T} & K_{t-1}^{(\tilde{y})} \end{pmatrix} \tag{7.30}$$

and $\mathring{k}_t = [k_t, \tilde{k}_t]^T$.

Proposition 24. *In the above setting, $\forall t \in \{1,\cdots,T\}$ and $\forall \mathbf{x}_t \in \mathscr{X}$, it holds that $\sigma_{t-1}(\mathbf{x}_t) \geq \mathring{\sigma}_{t-1}(\mathbf{x}_t)$.*

Proof. See Appendix for a detailed proof. □

7.5.4 Regret bounds

It is thus possible to apply the results from (Krause & Ong, 2011) to the case of $\tilde{f}$ and $\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$ and obtain the following regret bounds in the case where only the surrogate task is observed:

Proposition 25. *Let $\delta \in (0,1)$ and suppose that the following assumptions hold:*

1. *$\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$ is finite, $\tilde{f}$ is sampled from a known GP prior with known noise variance σ^2, and $\beta_t = 2\log\left(|\mathscr{X}|t^2\pi^2/(6\delta)\right)$.*

2. *$\mathscr{X} \subset [0,r]^d$ is compact and convex, $d \in \mathbb{N}$ and $r > 0$. Suppose that $\tilde{f}$ is sampled from a GP prior with known noise variance σ^2, and satisfies the following high probability bound on the derivations of GP sample paths $\tilde{f}$: for some constants $a, b > 0$*

$$\mathbb{P}\left(\sup_{\mathbf{x}\in\mathscr{X}} |\partial \tilde{f}/\partial x_m| > L\right) \leq a e^{-(L/b)^2}, \tag{7.31}$$

   for $m = 1, \cdots, d$, and pick $\beta_t = 2\log\left(t^2 2\pi^2/(3\delta)\right) + 2d\log\left(t^2 dbr\sqrt{\log(4da/\delta)}\right)$.

3. *$\mathscr{X}$ is arbitrary, $\|\tilde{f}\|_k \leq B$ and the noise variables ε_t form an arbitrary martingale difference sequence (i.e., $\mathbb{E}[\varepsilon_t|\varepsilon_1, \cdots, \varepsilon_{t-1}] = 0$, for all $t \in \mathbb{N}$), then set $\beta_t = 2B^2 + 300\gamma_t \log(t/\delta)^3$.*

The multi-objective contextual regret of CGP-UCB-MO then verifies

$$\mathbb{P}\left(\tilde{R}_T \leq \sqrt{C_1 T \beta_T \gamma_T} + 2\right) \geq 1 - \delta, \tag{7.32}$$

i.e., it is given by $\mathscr{O}^*(\sqrt{T\gamma_T\beta_T})$ with high probability.

Proof. This result lifts the results from Theorem 1 in (Krause & Ong, 2011). See the supplementary material for a detailed proof. □

Remark 28. Importantly, a careful analysis of (Krause & Ong, 2011)'s proofs shows that the results still hold under assumptions 1 and 2 in the presence of additional observations (the adaptation of assumption 3 if left for further research). Indeed, while $\overline{f} \in \tilde{\mathscr{X}}$, $\tilde{f}$ still belongs in $\mathscr{X}$ and has a known GP prior with known variance. Thus, under assumptions 1 or 2, given that the information gain γ_T is smaller with additional observations and the coefficient β_T remains the same, the regret $\tilde{R}_T$ is expected to be smaller, with high probability, than the regret of the standalone surrogate task.

7.6 Experimentation

In this section, the results of the proposed algorithm are demonstrated in a synthetic learning environment, where the optimal action is already known, and the regret metrics are compared across settings.

7.6.1 Data

A multi-objective reward function with 6 arms is considered, with the input space defined as $\mathscr{S} = \{\mathbf{s} \in \mathbb{R}^6_+ : \|\mathbf{s}\|_1 \leq 1\}$. 5 objectives are considered. The reward function for each objective is a simple function, such as $\sin(s), \cos(s), \tan(s), \log(s), as + b$ and b, for given constants a and b. Both $\tilde{f} = \mathbf{u}^T\mathbf{f}$ and a more complex task functional, namely the Gini Index function (Busa-Fekete et al., 2017), are considered. Hyper-parameters are chosen as follows: $\delta = 0.05$, β_t. An RBF kernel for $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{u}$ (with $\sigma = 1$), and the co-regionalisation kernel for the task are employed.

7.6.2 Results

CGP-UCB-MO comparison Figure 7.3 shows a mean cumulative regret comparison between the CGP-UCB-MO algorithm, and the mean and variance-only versions of these algorithms averaged over 50 repetitions. These alternatives are where the update rule in Equation 7.18 is replaced by either the first term (the mean, μ_t) or the second term (the standard deviation, σ_t).

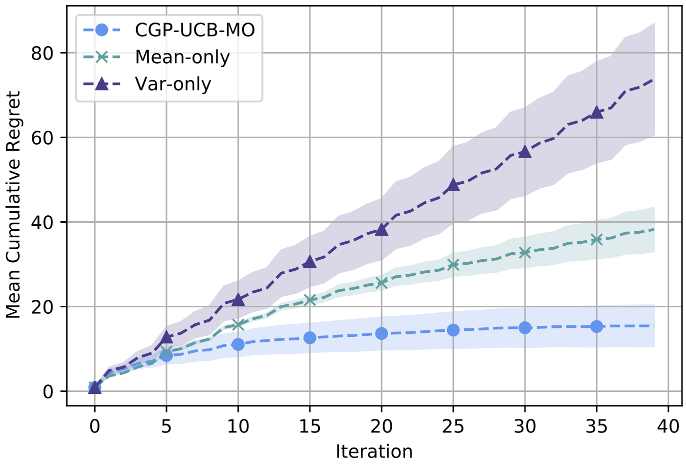

Figure 7.3: Comparing CGP-UCB-MO with a mean or variance only approach when choosing the next action. The plot shows the mean cumulative regret for each case in a single observation setting.

Observability comparison A comparison of different levels of observability is run and the performance when the surrogate reward function, or some extra information, is observed is monitored (7.4). In the full observation setting, at each iteration, the surrogate reward function and the reward function are observed for each objective separately. In the partial observation setting, the surrogate reward and also the reward for the first three single objectives once every 3 runs are observed. Finally, in the single observation setting, only the surrogate reward is observed. As can be seen from these results, more information gives a smaller regret over time, as the theoretical results indicate.

Benchmark comparison The proposed algorithm is benchmarked with that of (Busa-Fekete et al., 2017), using an implementation from (Zhacheny, 2019). The same synthetic setup as described in Section 8.2 of their paper is used. In the present case, consider $f^{[J+1]}(s) = -G_w(s)$, which corresponds

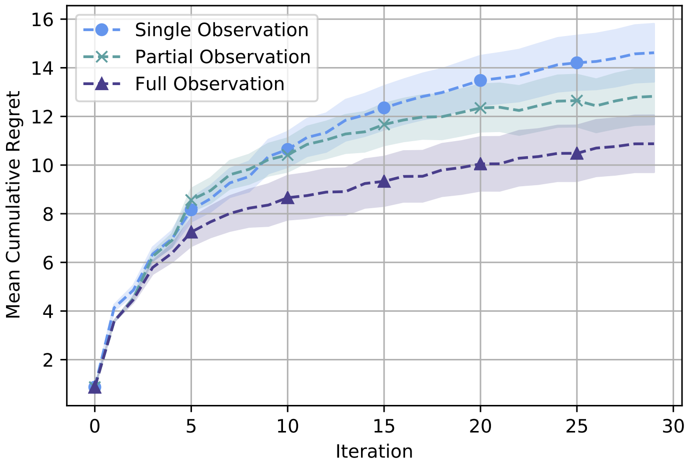

Figure 7.4: Comparing CGP-UCB-MO with different levels of observability using co-regionalisation. The plot shows the mean cumulative regret for each case.

to the same setting. In (Busa-Fekete et al., 2017), the authors consider $s_t = \arg\max_s -G_u(s\hat{u}_t)$, whilst, in the case of CGP-UCB-MO, Algorithm 4 is used. Furthermore, two approaches are adopted: a *smoothed* approach, where $\bar{\mathbf{s}}_{T+1} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{s}_t$, and an *unsmoothed* approach, where $\mathbf{s}_{T+1}$ is used. The results are shown in Figure 7.5, where performance is computed over 60 repetitions. The mean regret, which is calculated, at each time step, as the average of the regret across all simulations, is displayed. This is to match the work done in (Busa-Fekete et al., 2017). Note that in the present case, unlike in the original paper, there is no assumption that the Gini Index function is known, and the algorithm only samples from it, rather than using it in its optimisation.

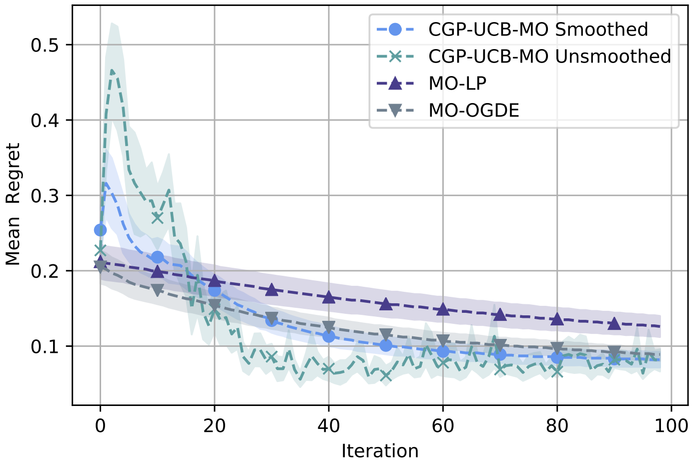

Figure 7.5: Comparing CGP-UCB-MO with MO-LP and MO-OGDE from (Busa-Fekete et al., 2017). The plot shows the mean regret for each case in a single observation setting.

7.7 Application to optimal execution

This Section now introduces an application of the proposed algorithm in quantitative finance.

7.7.1 The role of optimal execution in quantitative finance

Optimal execution is a critical concept in quantitative finance that refers to executing large trades in financial markets to minimise transaction costs and maximise profits for investors (Almgren & Chriss, 2001; Cartea et al., 2015; Crisafi & Macrina, 2014; Guéant, 2016). The execution of large trades can significantly impact market prices and, if executed inefficiently, can lead to significant losses for investors.

To understand optimal execution, it is essential first to introduce the concept of market impact. Market impact is the price change that occurs when a large trade is executed. Market impact is a function of various factors, including the size of the trade, the liquidity of the market, and the trading strategies used. The larger the trade, the greater the market impact. Optimal execution seeks to minimise the market impact by breaking up large trades into smaller trades that can be executed over time. This approach is known as a "slice and dice" approach. There are several approaches to optimal execution (Cartea et al., 2015; Guéant, 2016). One approach is known as the "time-weighted average price" (TWAP) strategy. This strategy involves executing trades at regular intervals over a specific time period. By executing trades at regular intervals, investors can achieve a price close to the average price over the entire trading period. Another approach is the "volume-weighted average price" (VWAP) strategy. This strategy involves executing trades in proportion to the trading volume over a specific time period. By executing trades in proportion to trading volume, investors can achieve a price close to the average price weighted by trading volume.

In addition to these strategies, other factors can impact optimal execution. One important factor is the choice of a trading venue. Different trading venues may have different levels of liquidity and transaction costs, which can impact execution prices. Another important input is the choice of the trading algorithm. Trading algorithms are computer programs that use mathematical models to execute trades automatically. Different algorithms may be better suited for different types of trades and market conditions, and choosing the right algorithm can significantly impact optimal execution.

7.7.2 Problem overview

Here, a reformulation of the optimal execution problem as a multi-objective problem is proposed, which is then solved using the proposed novel multi-armed bandit algorithm. Contrary to (Almgren & Chriss, 2001), optimal execution is considered here as a *multi-objective* problem, whereby an agent sells or buys a large amount of shares (thus maximising their profit), while minimising the adverse price movements that are a consequence of their own trades, over T steps. Indeed, the agent starts with an inventory Q_0 (to be liquidated) of a stock whose price is given by S_t at time t and chooses an amount a_t of shares to trade at each time-step, leading to a (running) cash profit X_t. The agent wishes to find an optimal strategy that trades off both objectives. Reformulating this as a multi-armed bandit

(MAB) problem (Cannelli et al., 2020)[6] enables the explicit inclusion of different risk appetites in the model.[7]

7.7.3 The Algrem-Chriss framework

In a continuous time setting, the optimal execution problem can be classically defined in terms of the following stochastic differential equations (Almgren & Chriss, 2001; Cartea et al., 2015)[8]:

$$\begin{aligned} dQ_t &= -a_t dt \\ dX_t &= (S_t - ka_t)\, a_t dt \\ dS_t &= -ba_t dt + \sigma dW_t, \end{aligned}$$

where a_t is the continuous time trading speed and $(W_t)_{t\geq 0}$ is a standard Brownian motion. k and b represent the temporary and permanent impact of a given a_t. As suggested in (Almgren & Chriss, 2001; Cartea et al., 2015), the agent is interested in maximising the expected return

$$\mathbb{E}\left[\underbrace{X_T + Q_T S_T}_{R_{0:T}} - \left(\underbrace{\alpha Q_T^2 + \phi \int_o^T Q_t^2 dt}_{C_{0:T}}\right)\right], \tag{7.33}$$

where $R_{0:T}$ can be thought of as the expected reward up until time T and $C_{0:T}$ can be considered as the inventory risk. The continuous-time formulation in Cartea et al., 2015 is discretised in the following.

7.7.4 Objectives

The *expected return*, $\mathbb{E}[R_{0:T}]$, to be maximised, consists of the proceeds of the liquidation and the present value of the remaining inventory at the time horizon T: $R_{0:T} := X_T + Q_T S_T$. The *expected cost*, $\mathbb{E}[C_{0:T}]$, to be minimised, penalises a non-zero inventory, both at T and intermediate steps t: $C_{0:T} := \alpha Q_T^2 + \phi \sum_{t=0}^{T-1} Q_t^2$, with $\phi > 0$.

7.7.5 Model dynamics

At each time-step $t = 0, \cdots, T-1$, the change in quantities of interest is given by:

$$Q_{t+1} - Q_t := \Delta Q_t = -a_t \tag{7.34}$$

$$X_{t+1} - X_t := \Delta X_t = (S_t + f(a_t))a_t \tag{7.35}$$

$$S_{t+1} - S_t := \Delta S_t = g(a_t) + \sigma \varepsilon_{t+1}, \tag{7.36}$$

[6] This is considered as a MAB problem as this is an *online learning* algorithm, which is important for trading during the course of the day. (Cannelli et al., 2020) also shows that bandits can outperform RL in practice.

[7] The views expressed therein are solely those of the authors and do not reflect those of any institution or employer, past or present. The authors make no representation and warranty whatsoever and disclaim all liability for the completeness, accuracy or reliability of the information contained herein. This chapter is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

[8] Note that the initial formulation by Algrem and Chriss was in discrete time.

where a_t is the discrete time trading speed, ε_{t+1} is the shock associated with the stock price, and b, k, σ are hyper-parameters. Here f is the temporary price impact function, whilst g is the permanent price impact function. with $k, b, \sigma > 0$ model parameters representing, respectively, temporary, permanent impacts, and volatility risk; ε_{t+1} represents a zero-mean random shock happening between times t and $t+1$ (note they need not be i.i.d.).

In the simplest settings, we often consider the $f(a_t) = -ka_t$, and $g(a_t) = -ba_t$ to be linear. Given that the functions f and g are usually unknown, expressing them as Gaussian processes is beneficial. We can reformulate this problem as follows:

$$\begin{bmatrix} \Delta Q_t \\ \Delta X_t \\ \Delta S_t \end{bmatrix} = \begin{bmatrix} -a_t \\ S_t a_t \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ a_t & 0 & 0 \\ 0 & 1 & \sigma \end{bmatrix} \begin{bmatrix} f_{t+1} \\ g_{t+1} \\ \varepsilon_{t+1} \end{bmatrix}. \tag{7.37}$$

Now, $\mathbf{y}_{t+1} := [f_{t+1}, g_{t+1}, \varepsilon_{t+1}]^T$ is a multi-output GP Bonilla et al., 2007.

7.7.6 Temporal credit assignment

A critical step in applying the proposed multi-armed bandit approach is to design step-wise rewards and costs rather than focus on terminal values. To do so, it is necessary to express values at the terminal time as $R_{0:T} = \sum_{t=0}^{T-1} r_t$ and $C_{0:T} = \sum_{t=0}^{T-1} c_t$, where

$$\begin{aligned} r_t &= (b-k)\,a_t^2 - ba_tQ_t + \sigma\,(Q_t - a_t)\,\varepsilon_{t+1} \\ c_t &= a_t^2 - 2Q_ta_t + \phi Q_t^2. \end{aligned}$$

Remark 29. These rewards and costs are known if underlying impact parameters b and k are available, which is not the case in practice.

7.7.7 Bandits for optimal execution

The expected return $R_{0:T}$ is traded off against market risk $C_{0:T}$ through multi-objective MABs. One can thus account for the user's risk appetite through the latter's preferences.

Single step vs multi step In the *single step* approach, rewards are sequentially observed at each discrete time step. In the *multi-step* approach, global optimisation is considered over all time steps up until time T. Table 7.1 compares the approaches, and Algorithm 5 shows the single-step approach. In both cases, a surrogate reward function of the form $\tilde{f}(x_t, u_t) = f(x_t)^T u_t$ is considered in the interest of simplicity. For each day, the *cumulative reward*, $R^{[d]}$, across day d, is also considered for comparison's sake.

Approach	Single Step	Multi-Step
Action Space	$\{[a_t]\}$	$\{[a_1, \cdots, a_T]\}$
Reward Function $f : \mathscr{X} \to \mathbb{R}^2$	$\mathscr{X} = \mathbb{R} \times Z \times U$ $f(x_t) = [r_t, -c_t]^T$	$\mathscr{X} = \mathbb{R}^T \times Z \times U$ $f(x_t) = [R_{0:T}, -C_{0:T}]$
Cumulative Reward	$\sum_{t=5(d-1)}^{5d} \tilde{f}(x_t, u_t)$	$\tilde{f}(x_d, u_d)$

Table 7.1: Single-Step vs Multi-Step Approach

Algorithm 5 Optimal Execution Single Step Algorithm

```
Require: $Q_0$, $S_0$, context $\mathbf{z_t}$, preference vector $\mathbf{u}_t$, Discrete time intervals per day $T$.
    for $t = 1, \cdots, T$ do
        Run CGP-UCB-MO to find action $a_t$ and observe surrogate reward $\tilde{f}(a_t, u_t)$ subject to $a_t < Q_t$
        Update $Q_{t+1} = Q_t - a_t$
    end for
    Calculate $R^{[d]}$ based on Table 7.1
    Reset $Q_t$ to $Q_0$ and $S_t$ to $S_0$
```

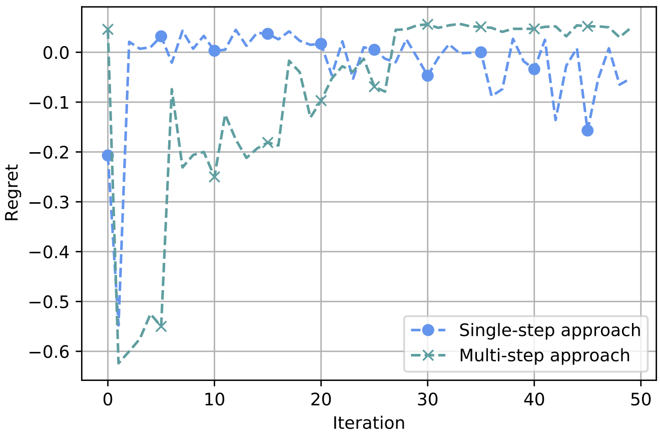

Figure 7.6: Reward over time for $u = [0.75, 0.25]$

7.7.8 Experimentation

Five discrete time steps are considered in a day ($T = 5$ in the multi-step approach). Each day, $Q_0 = 1$, $S_0 = 1,000$. Different preference vectors are chosen, but parameters are fixed ($b = 0.1, k = 1, \alpha = 1, \phi = 0.1$). Figure 7.6 shows the change in $R^{[d]}$ for each approach ($u = [0.75, 0.25]$). The multi-step approach reaches a better solution but is slower; the single step gets close to the closed-form solution, but is a little more unstable. Table 7.2 also shows this with different preference vectors. Note that we consider a standard radial basis function kernel in these experiments and do not vary the kernel choice, which is left for future research.

7.7.9 Further experiments

Here, additional plots for different preference vectors are presented, and comparisons with a closed-form solution are proposed.

Preference	$[1,0]$	$[0.75,0.25]$	$[0.5,0.5]$	$[0.25,0.75]$	$[0,1]$
Mean	0.202	0.095	0.088	0.055	-0.039
Median	0.222	0.088	0.064	0.072	-0.024

Table 7.2: Mean reward difference between multi-step and single-step approaches over 10 days.

7.7.9.1 Varying the preference vector

In this Subsection, additional preference scenarios are shown to illustrate the performance of single- and multi-step MABs.

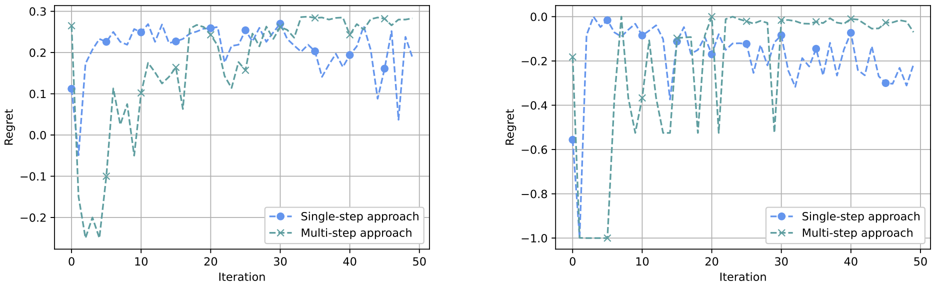

Figure 7.7: (a) Reward over time for $u = [0.5, 0.5]$, and (b) Reward over time for $u = [1, 0]$.

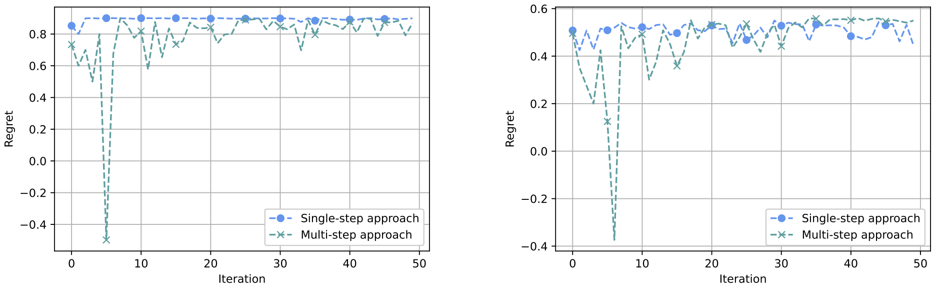

Figure 7.8: (a) Reward over time for $u = [0, 1]$, and (b) Reward over time for $u = [0.25, 0.75]$.

7.7.9.2 Closed-form expression

In the case where 1) the random variables ε_t $(t = 1, \cdots, T)$ are independent standard Gaussian variables and 2) the user's preferences are linear and static, the discrete optimal execution problem can be seen as the discretisation (or Euler-Maruyama scheme) of the usual continuous-time setting (Cartea et al., 2015), which corresponds to a model-based continuous-time reinforcement learning approach.

One can thus define a closed-form expression for μ_t in Equations 7.33, 7.33 and 7.33. In order to do this, firstly define

$$\gamma := \sqrt{\frac{\phi}{k}} \quad \text{and} \quad \psi := \frac{1 - 0.5b + \sqrt{k\phi}}{1 - 0.5b - \sqrt{k\phi}}. \tag{7.38}$$

The closed-form expression for the control v_t is given by (Cartea et al., 2015) as

$$v_t^* = \gamma \frac{\psi e^{\gamma \Delta_t (T-t)} + e^{-\gamma \Delta_t (T-t)}}{\psi e^{\gamma \Delta_t (T-t)} - e^{-\gamma \Delta_t (T-t)}} Q_t. \tag{7.39}$$

The optimal action is then simply computed as $a_t^* = v_t^* \Delta_t$, where Δ_t is the discretisation step.

The performance of the proposed algorithm can be benchmarked against that given by the closed-form solution. To do this, the case where the preference vector is $[1,1]$ is presented, making the reward function comparable to the closed-form expression setting. On each day, the quantity $\|a_t^* - a_t\|_1$ is computed. The results can be seen in Figure 7.9 (a). One can see that the multi-step approach obtains a near-optimal solution. The reward function including the closed-form solution is plotted in Figure 7.9 (b).

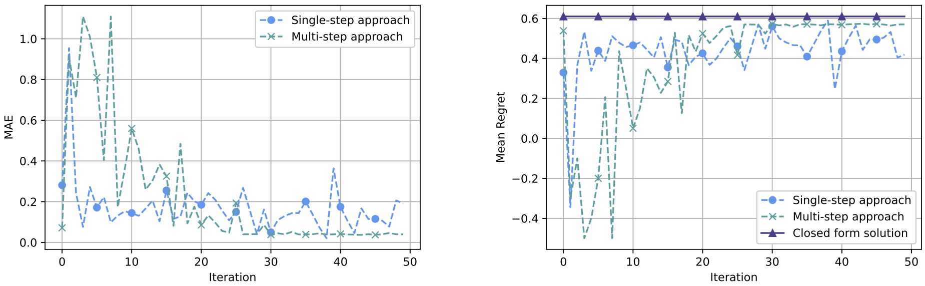

Figure 7.9: (a) Comparison of the approaches against the closed-form solution, and (b) Reward over time including the closed-form solution.

Remark 30. Importantly, Figure 7.9 (b) justifies the use of multi-armed bandits as (1) MABs are an online learning algorithm, which (2) is very close to the optimal solution obtained via model-based reinforcement learning. In real-life applications, as illustrated in a different use case by (Cannelli et al., 2020), the lack of independence across random noises, the violation of the Markov property, as well as regime shifts, make MABs more robust than RL.

7.8 Conclusion

In this chapter, an algorithm for learning multi-objective multi-armed bandit problems using Gaussian processes has been introduced. Firstly, the approach presented allows learning across the space of user preferences and accommodates varying levels of observability of the reward across the different objectives. Regret bounds for the problem have also been derived, and it has been shown that the addition of extra information leads to a decrease in information gain. This result is intuitive, as better learning should occur when more information is provided. Furthermore, it has been demonstrated, through experimentation, that increased information accelerates learning and that the input of preferences into the Gaussian process expedites the learning process.

Chapter 8

General Conclusions

This thesis has addressed a broad class of problems related to improving robustness in machine learning and, more generally, in artificial intelligence.

The first problem tackled in this thesis was the open question of the connection between margin maximisation, which is at the heart of most well-known classification algorithms, and thin tails. The work showed that margin maximisation techniques are related to a criterion guaranteeing that the underlying idiosyncratic "noise" random variables do not exhibit slowly-varying behaviour. This means that margin maximisation fundamentally relies on the assumption that error terms in the underlying dataset cannot produce rare events. The work demonstrated theoretically and empirically that this assumption should sometimes be relaxed for improved performance. However, further research is necessary to derive similar results in more complex settings, including multi-class labels and non-separable cases.

The second challenge was designing a flexible and efficient model for recommender systems. The work introduced the concept of "kernel factorisation machines," which are a generalisation of the usual factorisation machines. These machines can handle small- and big-data regimes while maintaining state-of-the-art performance with fewer parameters. Due to the curse of dimensionality, much of the work focuses on reducing the computational burden by carefully choosing inducing points as inputs.

The third issue addressed in this thesis was understanding the generalisation properties of debiasing algorithms in machine learning. In many cases, it is necessary to employ techniques to ensure that biases, particularly those related to protected attributes, are not present in a model's outputs. However, debiasing can lead to loss functions that are more complex and lower out-of-sample performance. The work derived bounds on the performance of debiasing algorithms (in terms of accuracy and fairness) on unseen data. These findings pave the way for more research into the parametrisation of partial debiasing depending on a specific use case and the amount of available data; future developments could also tackle other definitions of fairness, such as individual fairness.

This led to the fourth challenge, defining a generic framework for group fairness in unsupervised learning. The work considered the class of generalised low-rank models. It used so-called "group functionals" to establish a relationship between the expected loss observed amongst different categories, a generalisation of the usual empirical minimisation framework in machine learning. This simple setting enables practitioners to control the trade-off between performance and fairness straightforwardly and robustly. Results across multiple unsupervised learning tasks and datasets point to state-of-the-art performance. Further work could look at non-linear unsupervised learning and dimensionality reduction techniques and consider the practical impact of choosing a specific fairness metric in end-to-end systems.

The final part of the thesis explores multi-objective optimisation in machine learning, which is crucial for real-life situations where multiple (often conflicting) objectives must be considered. The research proposed new reinforcement learning and multi-armed bandit algorithms, enabling agents to learn policies over trade-offs. The work also presents a new perspective whereby an agent's preferences, such as a utility function with multiple inputs, can change over time. The proposed algorithms and perspective contribute to implementing multi-objective optimisation in machine learning. A key element that deserves more attention is the specification of preference dynamics.

Overall, this thesis has contributed to the understanding and improvement of the robustness of machine learning and artificial intelligence. It is hoped that the insights and techniques presented here will aid in developing more reliable and efficient AI systems with potential applications in a wide range of fields.

Appendix A

Fair Generalised Low-Rank Models

This appendix contains:

1. The definition of supervised fGLRMs

2. Omitted proofs of Propositions 1 and 2;

3. The proof of convergence of (some) algorithms fitting fGLRMs (alternating minimisation and biconvex search);

4. Empirical results of supervised GLRMs;

5. Empirical results of fair outcome-based fGLRMs, contrasted with fair cost-based results, demonstrating the existence of fairness-fairness trade-offs in unsupervised learning empirically.

A.1 Supervised fair GLRMs

A straightforward way of designing a “fair” unsupervised learning algorithm is proposed by introducing a pre-processing step that removes “unfair” features. This is similar to the algorithm proposed in (Wang & Wang, 2021). Instead of using the $A_{i,j}$’s directly, an auxiliary matrix $\tilde{A}$ is considered, which consists of features that are (to some extent) unrelated to the protected characteristic s.

Using the example of SVD or matrix factorisation, columns of the original matrix corresponding to features that are highly correlated to the protected characteristic are removed. In the case of k-means, this boils down to ignoring some of the features when computing distances between data points. This can be seen as a straightforward generalisation of supervised principal components (see Section 18.6 and Algorithm 18.1 in (Hastie et al., 2009))

Algorithm 6 Naive Supervised Fair Generalised Low Rank Model

```
Require: Initial features $\mathbf{f}$, protected attribute $s$ and threshold $\theta$.
  Set $\mathscr{F} = \varnothing$
  for $m = 1, \cdots, M$ do
    Compute the univariate logistic regression $f_m$ on the protected characteristic $s$
    $\theta_m \leftarrow \mathrm{AUC}_m$
    if $\theta_m < \theta$ then
      $\mathscr{F} \leftarrow \mathscr{F} \cup \{f_m\}$
    end if
  end for
  Perform GLRM on the reduced set of features $\mathscr{F}$ and calculate $\mathbf{X}^t, \mathbf{Y}^t$
  return $\mathbf{X}^t, \mathbf{Y}^t$
```

Here, the area under the curve (“AUC”) was chosen as a measure of fit, and thus correspondence, between a given feature f_m and s, but other metrics can obviously be used Any feature with a high AUC thus gets removed. A standard GLRM model is then run on the remaining features. The drawback of this approach, however, is that part of the data is ignored. Empirical results can be found in the Appendix.

A.2 Proofs

The proofs of Propositions 1 and 2 are presented here.

A.2.1 Proof of Proposition 1

Proof. The proof adapts results from Example 3.14 in (Boyd & Vandenberghe, 2004) to the case of wLSE. First, recall the definition of weighted Log-Sum-Exponential:

Definition 15. The weighted Log-Sum-Exponential (“wLSE”) is defined as

$$T(z_1, \cdots, z_K) = \frac{1}{\alpha} \log \left(\sum_{k=1}^{K} w_k \, e^{\alpha z_k} \right), \tag{A.1}$$

Convexity. For the sake of simplicity, suppose that every w_k is (strictly) positive (otherwise, the corresponding index can simply be removed from the wLSE). Now, each function $h_k : z_k \mapsto w_k e^{\alpha z_k}$ is *log-convex* (Boyd & Vandenberghe, 2004) since $\log(h_k(z_k)) = \log w_k + \alpha z_k$ is convex in $(z_1, \cdots, z_K)$. Log-convexity is preserved under sums, so that $\sum_{k=1}^{K} w_k \, e^{\alpha z_k}$ is log-convex, hence the result.

Shift property. It is straightforward to check that

$$\begin{aligned} T(z_1, \cdots, z_K) &= \frac{\log(e^{\alpha \bar{z}} \sum_{k=1}^{K} w_k \, e^{\alpha(z_k - \bar{z})})}{\alpha} && \text{(A.2)} \\ &= \bar{z} + \frac{\log \sum_{k=1}^{K} w_k \, e^{\alpha(z_k - \bar{z})}}{\alpha} && \text{(A.3)} \\ &= \bar{z} + T(z_1 - \bar{z}, \cdots, z_K - \bar{z}). && \text{(A.4)} \end{aligned}$$

Limiting cases. Case $\alpha \to 0$:

$$\begin{aligned}
\lim_{\alpha \to 0} T(z_1, \cdots, z_K) &= \lim_{\alpha \to 0} \frac{\log(\sum_{k=1}^K w_k\, e^{\alpha z_k})}{\alpha} && \text{(A.5)}\\
&= \lim_{\alpha \to 0} \frac{\frac{\sum_{k=1}^K w_k z_k\, e^{\alpha z_k}}{\sum_{k=1}^K w_k\, e^{\alpha z_k}}}{1} && \text{(A.6)}\\
&= \sum_{k=1}^K w_k z_k, && \text{(A.7)}
\end{aligned}$$

where the second line comes from L'Hôpital's rule.

Case $\alpha \to +\infty$:

$$\begin{aligned}
T(z_1, \cdots, z_K) &= \frac{\log(e^{\alpha \max z_k} \sum_{k=1}^K w_k\, e^{\alpha(z_k - \max z_k)})}{\alpha} && \text{(A.8)}\\
&= \max z_k + \frac{\log \sum_{k=1}^K w_k\, e^{\alpha(z_k - \max z_k)}}{\alpha}, && \text{(A.9)}\\
& && \text{(A.10)}
\end{aligned}$$

and the second term goes to zero as $\alpha \to +\infty$.

□

A.2.2 Proof of Proposition 2

Proof. The proof is based on the repeated use of results on the composition of convex functions, see Section 3.2.4 in (Boyd & Vandenberghe, 2004). First, observe that for each (i, j), $\ell_{i,j}$ is biconvex in X and Y since it is a function of x_i and y_j only and is biconvex in those. By summing convex functions, this implies that each z_k is biconvex in X and Y. Second, suppose that T is convex and non-decreasing in each argument (this is trivially the case for the wLSE and all fairness functionals considered here). Therefore, T composed by a convex function is still convex (see Equation (3.15) in (Boyd & Vandenberghe, 2004) and thereafter). But $z(\cdot, Y) = (z_1(\cdot, Y), \cdots, z_K(\cdot, Y))$ is convex in X (and $z(X, \cdot)$ in Y). It thus preserves the convexity in X (keeping Y fixed) and in Y (keeping X fixed), hence the biconvexity in X and Y. □

A.3 Algorithms

In this Section, some algorithms' convergence properties are discussed, in particular alternating minimisation and biconvex search. The reader is referred to (Boyd & Vandenberghe, 2004; Gorski et al., 2007; Hastie et al., 2015b) for in-depth treatment of these algorithms.

A.3.1 Alternating Minimisation

If each inner minimisation problem is solvable then the overall loss function $\mathscr{L}$ decreases at each iteration t. Now, if $\mathscr{L}$ is bounded by below (which is implied by the existence of a lower bound on the individual loss functions $\ell_{i,j}$), it can be deduced that the sequence $\{\mathscr{L}(\mathbf{X}_t, \mathbf{Y}_t)\}_{t=1,2,\cdots}$ generated by Algorithm 1 converges monotonically. This is similar to Theorem 4.5 in (Gorski et al., 2007).

A.3.2 Biconvex search

Here, the previous algorithm is adapted to optimise the objective function block by block, as opposed to coordinate by coordinate and the entire objective is required to be biconvex (which is guaranteed by the convexity of each $\ell_{i,j}$).

Algorithm 7 Biconvex Search

```
Require: Matrix A, loss functions $\ell_{i,j}$ and penalty functions $r_i$ and $\tilde{r}_j$.
    Select initial values $\mathbf{X}^0$ and $\mathbf{Y}^0$
    $t \leftarrow 0$
    repeat
        for $i = 1, \cdots, n$ do
            $\mathbf{X}_{t+1} \leftarrow \arg\min_{\mathbf{X}} T(z(\mathbf{X}, \mathbf{Y}_t)) + \sum_{i=1}^{n} r_i(x_i)$
        end for
        for $j = 1, \cdots, p$ do
            $\mathbf{Y}_{t+1} \leftarrow \arg\min_{\mathbf{Y}} T(z(\mathbf{X}_{t+1}, \mathbf{Y})) + \sum_{j=1}^{p} \tilde{r}_j(y_j)$
        end for
        $t \leftarrow t + 1$
    until convergence
    return $\mathbf{X}_t, \mathbf{Y}_t$
```

Since biconvex search can be seen as a particular case of alternating minimisation, the previous result also applies to the sequence generated by this algorithm. However, under certain circumstances, additional convergence properties are available on the parameters $\mathbf{X}$ and $\mathbf{Y}$ themselves, as opposed to $\mathscr{L}(\mathbf{X}, \mathbf{Y})$. Now, suppose that $\mathbf{X} \in \mathscr{X} \subseteq \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathscr{Y} \subseteq \mathbb{R}^{p \times d}$, where d is the chosen dimension of each x_i and y_j. If $\mathscr{X}$ and $\mathscr{Y}$ are closed and $\mathscr{L}$ is continuous, then, by a simple adaptation of Theorem 4.7 in (Gorski et al., 2007), if $(\mathbf{X}_t, \mathbf{Y}_t)$ converges to $(\mathbf{X}^*, \mathbf{Y}^*)$, then $(\mathbf{X}^*, \mathbf{Y}^*)$ is a partial optimum of $\mathscr{L}$.

Importantly, more precise convergence properties will depend on the specific problem at hand (e.g., k-means, PCA, non-negative matrix factorisation, etc.), but the broad convergence properties mentioned here are usually sufficient in practice.

A.3.3 Biconvex gradient descent

Finally, the focus turns to biconvex gradient descent, in the case where the $\ell_{i,j}$'s, r_i's and $\tilde{r}_j$'s do not take infinite values and are convex (if they are not differentiable, one can take a subgradient instead).

This can be understood as simply taking one step in biconvex search, as opposed to solving the entire minimisation problem for each block. If the step size is small enough, then the loss will decrease at each iteration. However, this is not guaranteed in general and one should apply back-tracking. When this is the case, and assuming that the loss is bounded by below, then the latter converges monotonically. Additional discussion can also be found in (Udell et al., 2016).

A.4 Additional experiments

A.4.1 Varying weighting scheme and functional

Here, results demonstrating the flexibility of the framework are introduced, showing its ability to use different weighting schemes, as well as underlying functionals. In Figure A.1 we show a comparison of q-FFL proposed in (Li et al., 2020b) with wLSE functional. Results are similar, both converging to the min-max solution for large parameters α and q respectively.

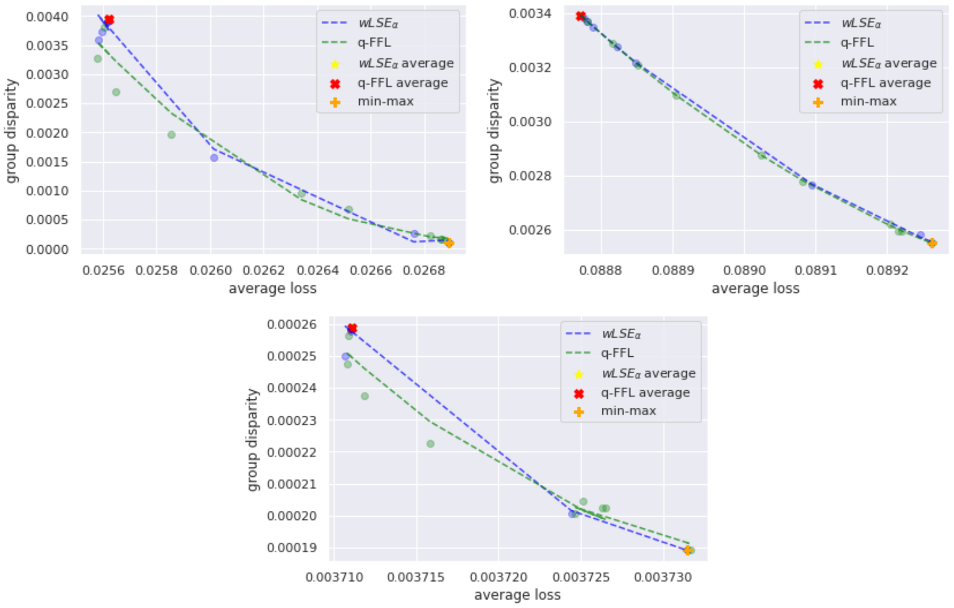

Figure A.1: NMF. Average loss and group disparity trade-off curves with the LSE and q-FFL functionals. Each dot corresponds to a different value of α, on the Adult, German Credit, and Loan Defaults datasets.

Average loss and group disparity trade-off curves are also compared when the weighting scheme is changed from empirical weights (i.e., weights are determined by the number of instances in each protected group observed in the data) to uniform weights (i.e., each protected group receives the same weight) in Figure A.2. Again it can be observed that, for large values of α, both curves converge to the min-max solution. More interestingly, using uniform weights can lead to a lower group disparity on average. However, the results are slightly mixed and likely to be data-dependent.

A.4.2 Supervised GLRMs

Here, a comparison of absolute differences between group costs obtained by supervised GLRM and standard approaches on German Credit (Dua & Graff, 2017), Loan Defaults (Yeh & hui Lien, 2009) and Adult (Dua & Graff, 2017) datasets is proposed. The comparison for both PCA (Figure A.3) and k-means (Figure A.4) is presented. The threshold θ used to decide which features will be removed was selected after reviewing the range of AUC values at the data set level. In the case of the German Credit, Loan Defaults and Adult data sets θ was set to 53%, 51%, and 65% respectively, which led to the removal of 8, 15, and 4 features respectively.

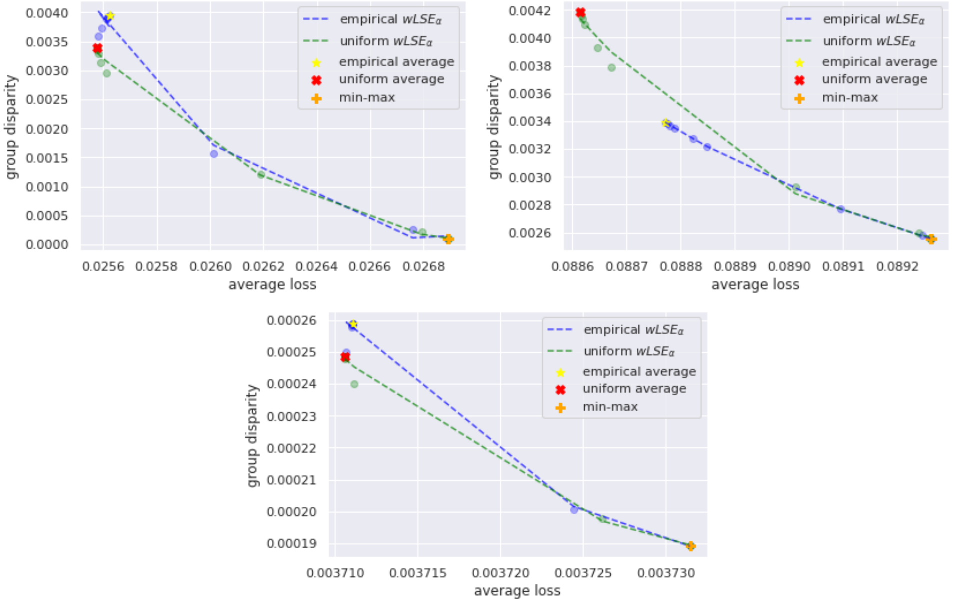

Figure A.2: NMF. Average loss and group disparity trade-off curves with empirical and uniform protected group weights. Each dot corresponds to a different value of α, on Adult, German Credit, and Loan Defaults datasets.

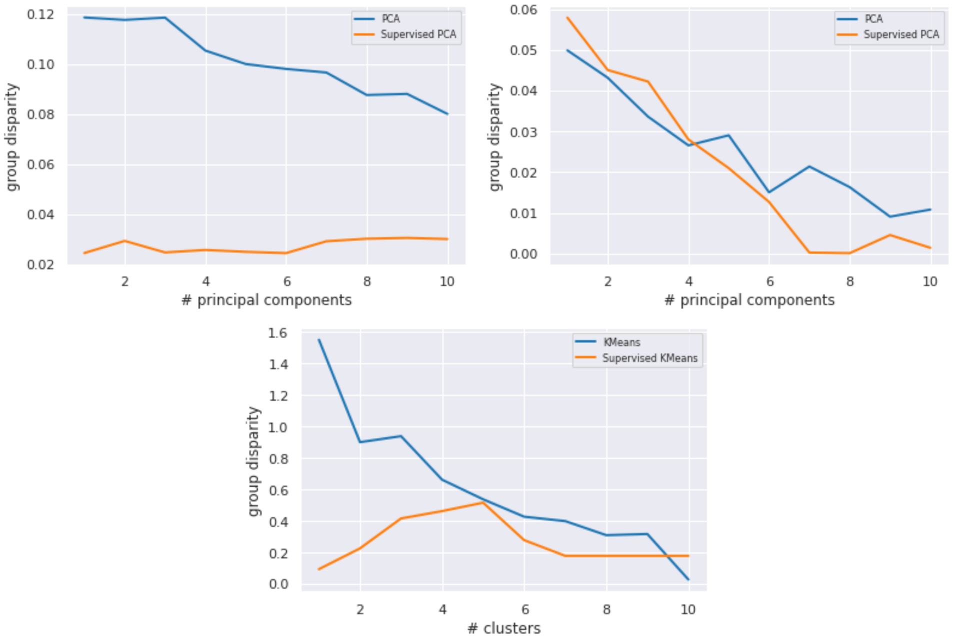

Figure A.3: PCA. Comparison of group disparity (i.e., absolute difference of costs incurred by different protected groups) on adult, German credit, and loan defaults data sets.

A.4.3 Outcome-based fGLRMs

In this Section, the *cost-based* fairness with *outcome-based* fairness.

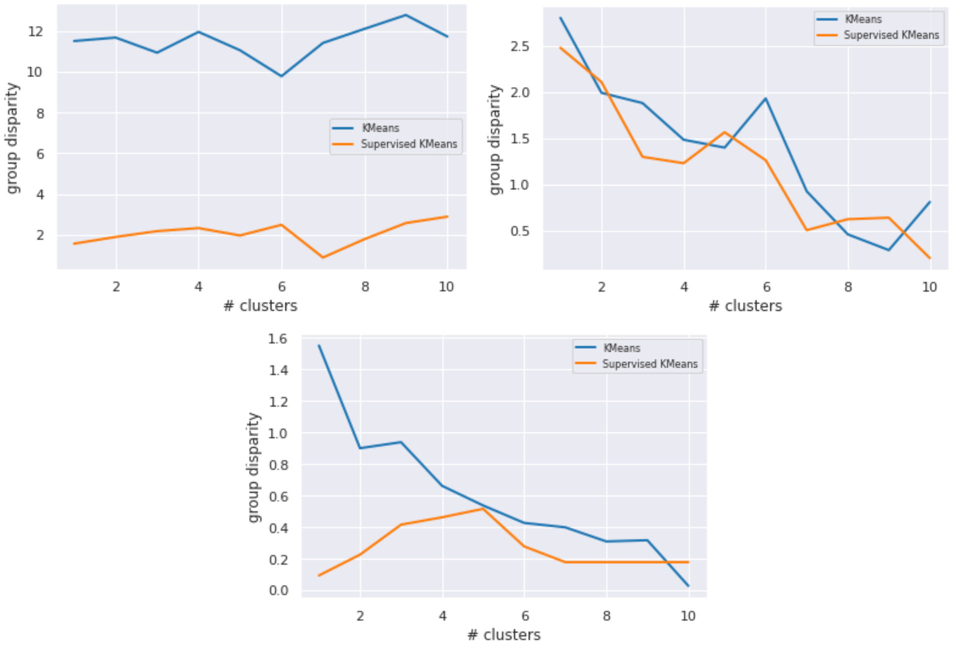

Figure A.4: KMeans. Comparison of group disparity (i.e., absolute difference of costs incurred by different protected groups) on adult, German credit, and loan defaults data sets.

A.4.3.1 A new penalty term

One can understand the prediction $\widehat{A}_{i,j} = x_i \cdot y_j$ as the *outcome* of the unsupervised learning algorithm. In line with, say *demographic parity* (Narayanan, 2018), one may wish to ensure that $\widehat{A}_{\cdot,j}$ is independent of the protected attribute s for all $j = 1, \cdots, p$: $\widehat{A}_{\cdot,j} s.$. It is usual, however, to replace the independence requirement by a zero (empirical) covariance constraint (Zafar et al., 2017b), i.e., for each $j = 1, \cdots, p$, we wish to have (where $\bar{s}$ is the average protected characteristic)

$$\frac{1}{n}\sum_{i=1}^{n} \widehat{A}_{i,j}(s_i - \bar{s}) = 0, \tag{A.11}$$

which can be rewritten as

$$y_j \cdot \left[\frac{1}{n}\sum_{i=1}^{n} x_i(s_i - \bar{s})\right] = 0. \tag{A.12}$$

Now, introducing the notation $\bar{x}^s = \frac{1}{n}\sum_{i=1}^{n} x_i(s_i - \bar{s})$, this is akin to adding the bilinear constraints: $y_j \cdot \bar{x}^s = 0$ for each j. These can be directly added in the GLRM (Equation 5.1) or in the fGLRM programme (Equation 5.4), or turned into soft constraints as in (Zafar et al., 2017b). Here, however, we choose a slightly different route in the spirit of GLRMs and add a penalty to the objective function, similarly to the barrier method (Boyd & Vandenberghe, 2004). For each j, we thus introduce a function ψ_j to penalise any deviation, such as $\psi_j(u) = \gamma_j u^p$ (where $p \geq 1$ and $\gamma_j \geq 0$), and introduce a modified fGLRM:

$$\mathscr{L}(X,Y) = T\left(z_1(X,Y),\cdots,z_K(X,Y)\right) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{p} \tilde{r}_j(y_j) + \sum_{j=1}^{p} \psi_j\left(\bar{x}^s \cdot y_j\right). \tag{A.13}$$

Importantly, this new penalty term depends on both x and y and is itself, under the assumption of ψ_j's convexity, biconvex.

A.4.3.2 Results

In our experiments, we used a quadratic penalty function for each feature (i.e., ψ_j) and we also used a weight ν that varies the contribution of orthogonality constraints to the overall cost function. This led to the following additional term

$$\psi_j(\bar{x}^s \cdot y_j) := \nu \left[y_j^T \left[\frac{1}{n} \sum_{i=1}^{n} x_i(s_i - \bar{s}) \right] \right]^2, \tag{A.14}$$

which we denote below as 'orthogonality metric'.

In Figure A.5, we show the evolution of the orthogonality metric for three different solutions - standard PCA, PCA with orthogonality constraint (i.e., $wLSE_\alpha$ with $\alpha = 10^{-4}$ and $\nu = 10^5$), and min-max (i.e., $wLSE_\alpha$ with $\alpha = 10^4$ and $\nu = 0$). As one can see from Figure A.5, the orthogonality metric tends to be the smallest in the approach that incorporates orthogonality constraint. One can also note that the orthogonality metric tends to be highest in the min-max approach, which is slightly unexpected.

Similarly, these results are compared to cost-based results shown in Figure A.6 and notice that the smallest group disparity is obtained using the min-max method. But, this time, PCA with orthogonality constraint leads to a higher group disparity.

In short, this shows that cost-based and outcome-based approaches are generally measuring different quantities, and are not necessarily consistent. To some extent, one can see these results as fairness-fairness trade-offs in the case of unsupervised learning, as increasing a given fairness metric may lead to decreasing another.

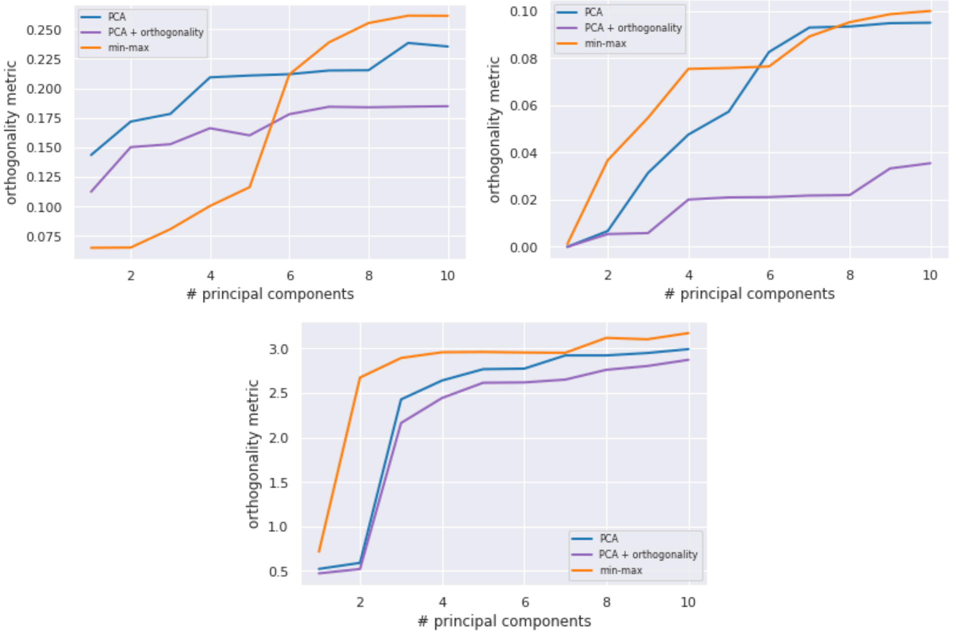

Figure A.5: Comparison of total squared orthogonality measure introduced in Equation A.14 for standard PCA, PCA with orthogonality constraint (using $wLSE_\alpha$ with $\alpha = 10^{-4}$ and $\nu = 10^5$), and min-max solutions (i.e., $wLSE_\alpha$ with $\alpha = 10^4$ and $\nu = 0$) on adult, credit, and LFW data sets.

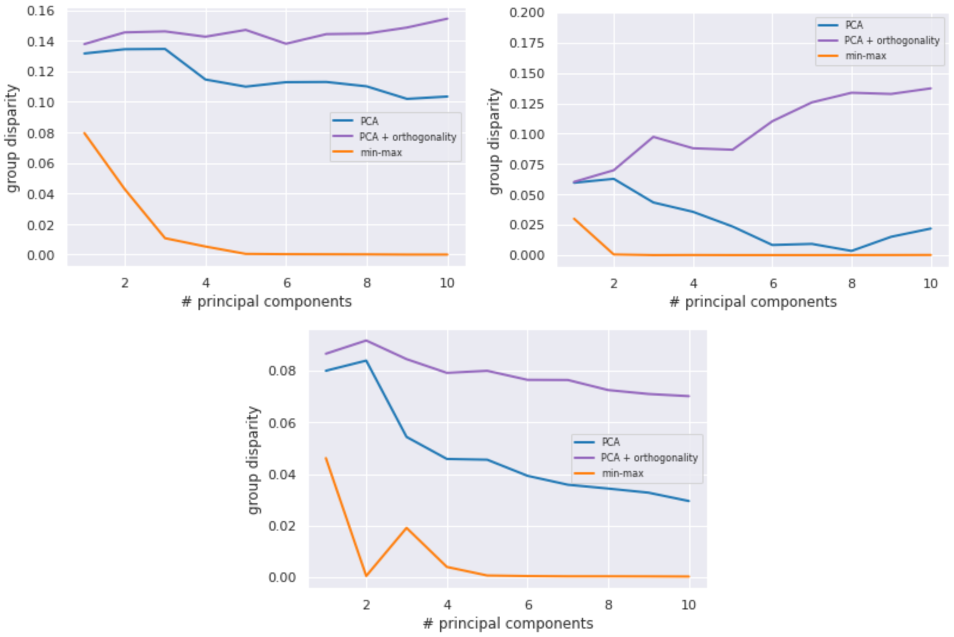

Figure A.6: Comparison of group disparity (i.e., absolute difference in approximation errors incurred by different protected groups) for standard PCA, PCA with orthogonality constraint (using $wLSE_\alpha$ with $\alpha = 10^{-4}$ and $\nu = 10^5$), and min-max solutions (i.e., $wLSE_\alpha$ with $\alpha = 10^4$ and $\nu = 0$).

Appendix B

Robust Multi-Objective Reinforcement Learning

B.1 Ablation Study

In this Section, an ablation study is provided, and the effects of exploration with exemplar models (Fu et al., 2017) and adaptive task weights using wLSE (Buet-Golfouse & Utyagulov, 2022c) are quantified. The ablation study is carried out by understanding the effects of exemplar exploration and wLSE on the model's performance.

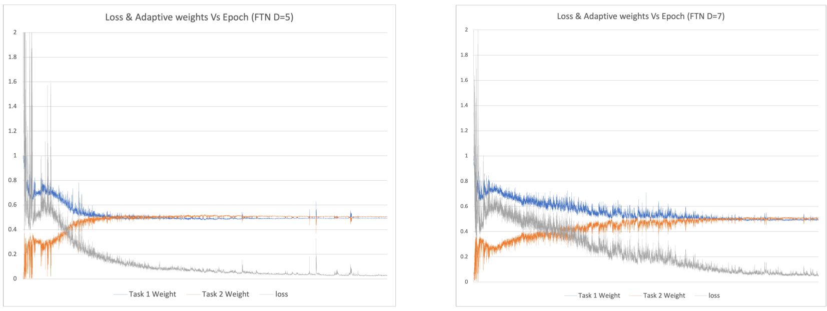

Figure B.1: Convergence plots for Fruit Tree Network under dynamic preferences setting, Task 1: multi-objective Q-network loss, Task 2: policy Q-network loss

The ability of the model to retrieve the CCS is evaluated on two environments, Fruit Tree and Deep Sea Treasure and consider four scenarios: (1) Replacing wLSE with weighted average while computing the loss of the overall network (without wLSE), (2) Removing exemplar exploration, (3) replacing wLSE with weighted average and using only ε-greedy exploration, (4) using both wLSE and exemplar exploration. The coverage ratio results are provided in Tables B.1-B.2 and the convergence plot under (4) in figure B.1. In environments with smaller state space, FTN ($depth = 5$) and fewer objectives, DST, there are no scalability requirements while in complex scenarios with high dimensionality surrogate state spaces, the performance suffers. This is evident in FTN ($depth =$ 6 and 7) where a boost in F1 score is apparent thanks to both efficient exploration and adaptive task weights.

Table B.1: Fruit Tree Network: The networks are trained on 3000 episodes and the coverage ratio is calculated on a random sample of 2000 preferences over 10 trials.

Tree Depth	Without wLSE	Without Exemplar	Without wLSE and Exemplar	with Both
5	1.0000 ±0.000	1.0000 ±0.000	1.0000 ±0.000	1.0000 ±0.000
6	0.9421 ±0.007	0.9201 ±0.005	0.9104 ±0.005	0.9912 ±0.009
7	0.7057 ±0.015	0.6181 ±0.010	0.7726 ±0.013	0.8326 ±0.020

Table B.2: Deep Sea Treasure: The networks are trained on 2000 episodes, and the coverage ratio is calculated on a random sample of 2000 preferences over 10 trials.

Without wLSE	Without Exemplar	Without wLSE and Exemplar	with Both
1.0000 ±0.000	1.0000 ±0.000	0.8720 ±0.002	1.0000 ±0.000

Appendix C

Multi-Objective Context Gaussian Process Upper Confidence Bound

In this Appendix, the following topics are covered:

- Section C.1: Proofs of theoretical results in the main text.
- Section C.2: Additional experimental results.

C.1 Proofs

In this section, we provide complete proofs of the theoretical results in the main text.

C.1.1 Proposition 23

Proposition. *If two kernels, k and $\mathring{k}$, are available with the same observations points, then the difference in information gain is:*

$$I(\mathbf{y}_T; f; k) - I(\mathbf{y}_T; f; \mathring{k}) = \frac{1}{2}\sum_{t=1}^{T} \log\left(\frac{1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)}{1+\sigma^{-2}\mathring{\sigma}_{t-1}^2(\mathbf{x}_t)}\right). \tag{C.1}$$

In particular, if $\mathring{\sigma}_{t-1}^2(\mathbf{x}_t) \leq \sigma_{t-1}^2(\mathbf{x}_t)$ for all t's, then $I(\mathbf{y}_T; f; \mathring{k}) \leq I(\mathbf{y}_T; f; k)$. Finally, this leads to

$$\gamma(T; \mathring{k}; \mathscr{X}) \leq \gamma(T; k; \mathscr{X}). \tag{C.2}$$

Proof. The expression for the information gain follows directly from (Krause & Ong, 2011). In particular, we notice that $\log\left(\frac{1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)}{1+\sigma^{-2}\mathring{\sigma}_{t-1}^2(\mathbf{x}_t)}\right) \geq 0$ if $\sigma_{t-1}^2(\mathbf{x}_t) \geq \mathring{\sigma}_{t-1}^2(\mathbf{x}_t)$. Thus, if $\sigma_{t-1}^2(\mathbf{x}_t) \geq \mathring{\sigma}_{t-1}^2(\mathbf{x}_t)$ holds for all $t = 1, \cdots, T$, the result follows. Now, since

$$\gamma(T; k; \mathscr{X}) := \max_{A \subset S: |A| = T} I(\mathbf{y}_A; f; k),$$

it comes

$$\gamma(T;\mathring{k};\mathscr{X}) \leq I(\mathbf{y}_{A^*};f;k) \leq \gamma(T;k;\mathscr{X}), \tag{C.3}$$

where $A^* = \arg\max_{A \subset S:|A|=T} I(\mathbf{y}_A;f;\mathring{k})$. □

C.1.2 Proposition 24

Firstly, note the following proposition.

Proposition 26. *For a block matrix* $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$*, we have that*

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} F & -FBD^{-1} \\ -D^{-1}CF & D^{-1}+D^{-1}CFBD^{-1} \end{pmatrix} \tag{C.4}$$

where $F = (A - BD^{-1}C)^{-1}$

Proof of Proposition 26. By the block matrix inversion lemma (Bernstein, 2009). □

Applying Proposition 26 to $\mathring{K}_t$ leads to the following result:

Proof of Proposition 24.

$$\begin{aligned} \mathring{k}_t^T \left(\mathring{K}_t + \sigma^2 I\right)^{-1} \mathring{k}_t = k_t^T (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} k_t - \\ \tilde{k}_t^T (K^{(\tilde{y})\,-1}_{t-1} C (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1}) k_t \\ - k_t^T (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} \tilde{k}_t \\ + \tilde{k}_t^T K^{(\tilde{y})\,-1}_{t-1} \tilde{k}_t + \tilde{k}_t^T (K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1} (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} \cdot K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1}) \tilde{k}_t \end{aligned}$$

Now, $\sigma^2(\mathbf{x}_t) = k(\mathbf{x}_t,\mathbf{x}_t) - \tilde{k}_t D^{-1} \tilde{k}_t$. Cancelling terms in the following expression leaves

$$\begin{aligned} \sigma^2(\mathbf{x}_t) - \mathring{\sigma}^2(\mathbf{x}_t) = k_t^T (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} k_t - \\ \tilde{k}_t^T (K^{(\tilde{y})\,-1}_{t-1} C (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1}) k_t \\ - k_t^T (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} \tilde{k}_t \\ + \tilde{k}_t^T (K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1} (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} \cdot K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1}) \tilde{k}_t. \end{aligned}$$

This can be further factorised to yield

$$\begin{aligned} \sigma^2(\mathbf{x}_t) - \mathring{\sigma}^2(\mathbf{x}_t) &= (k_t - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} \tilde{k}_t)^T (K^{\mathscr{O}}_{t-1} - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} K^{(\mathscr{O},\tilde{y})T}_{t-1})^{-1} \cdot (k_t - K^{(\mathscr{O},\tilde{y})}_{t-1} K^{(\tilde{y})\,-1}_{t-1} \tilde{k}_t) \\ &\geq 0, \end{aligned}$$

where the final inequality holds since $(K^{\mathcal{O}}_{t-1} - K^{(\mathcal{O},\tilde{y})}_{t-1} K^{(\tilde{y})}_{t-1}{}^{-1} K^{(\mathcal{O},\tilde{y})T}_{t-1})^{-1}$ is a positive semi-definite matrix, due to its being a Schur complement (Zhang, 2006). Thus, $\sigma(\mathbf{x}_t) \geq \mathring{\sigma}(\mathbf{x}_t)$. □

C.1.3 Proposition 25

Proposition. *Let $\delta \in (0,1)$ and suppose that the following assumptions hold:*

1. *$\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$ is finite, $\tilde{f}$ is sampled from a known GP prior with known noise variance σ^2, and $\beta_t = 2\log\left(|\mathscr{X}|t^2\pi^2/(6\delta)\right)$.*

2. *$\mathscr{X} \subset [0,r]^d$ is compact and convex, $d \in \mathbb{N}$ and $r > 0$. Suppose that $\tilde{f}$ is sampled from a GP prior with known noise variance σ^2, and satisfies the following high probability bound on the derivations of GP sample paths $\tilde{f}$: for some constants $a, b > 0$*

$$\mathbb{P}\left(\sup_{\mathbf{x}\in\mathscr{X}} |\partial \tilde{f}/\partial x_m| > L\right) \leq ae^{-(L/b)^2}, \tag{C.5}$$

for $m = 1, \cdots, d$, and pick $\beta_t = 2\log\left(t^2 2\pi^2/(3\delta)\right) + 2d\log\left(t^2 dbr\sqrt{\log(4da/\delta)}\right)$.

3. *$\mathscr{X}$ is arbitrary, $\|\tilde{f}\|_k \leq B$ and the noise variables ε_t form an arbitrary martingale difference sequence (i.e., $\mathbb{E}[\varepsilon_t|\varepsilon_1, \cdots, \varepsilon_{t-1}] = 0$, for all $t \in \mathbb{N}$), then set $\beta_t = 2B^2 + 300\gamma_t \log(t/\delta)^3$.*

The multi-objective contextual regret of CGP-UCB-MO then verifies

$$\mathbb{P}\left(\tilde{R}_T \leq \sqrt{C_1 T \beta_T \gamma_T} + 2\right) \geq 1 - \delta, \tag{C.6}$$

i.e., it is given by $\mathscr{O}^*(\sqrt{T\gamma_T\beta_T})$ with high probability.

Proof. There are two cases: the no additional case (i.e., only the surrogate task is observed) and a case where additional information is available.

In the first case, one can directly apply Theorem 1 in (Krause & Ong, 2011) to the space $\mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$, which is the same as concatenating the preference space $\mathscr{U}$ and the context space $\mathscr{Z}$ into a surrogate context space $\mathscr{Z}'$.

In the second case, it is not possible to directly apply their Theorem. The proof strategy, however, is simple and consists of systematically modifying the intermediate results in the supplemental material of (Krause & Ong, 2011), by replacing $\mathbf{z}_t$ with $\mathbf{z}'_t = (\mathbf{u}_t, \mathbf{z}_t)$.

Lemma 2. *Fix $t \geq 1$. If $|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X} = \mathscr{S} \times \mathscr{U} \times \mathscr{Z}$, then the multi-objective contextual regret is bounded by $2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_t)$.*

Proof. The proof is similar to the proof of Lemma 4.1 in (Krause & Ong, 2011), with $\mathbf{z}'_t = (\mathbf{u}_t, \mathbf{z}_t)$. □

Now, introduce a discretised space $S_t \subset \mathscr{S}$.

Lemma 3. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log|S_t|\pi_t/\delta$, where $\sum_{t\geq 1} 1/\pi_t = 1$, $\pi_t > 0$. Then*

$$|f(\mathbf{s},\mathbf{u}_t,\mathbf{z}_t) - \mu_{t-1}(\mathbf{s},\mathbf{u}_t,\mathbf{z}_t)| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{s},\mathbf{u}_t,\mathbf{z}_t), \tag{C.7}$$

for all $\mathbf{s} \in S_t$, for all $t \geq 1$, holds with probability $1-\delta$.

Proof. The proof applies directly Lemma 5.5 in (Srinivas et al., 2009) and Lemma 5.2 in (Krause & Ong, 2011). □

Importantly, this result utilises properties of Gaussian variables and uses parameters μ_{t-1} and σ_{t-1} obtained with *all the information* available up to step $t-1$.

Further, recall Lemmas 5.3 and 5.4 in (Krause & Ong, 2011), using $\mathbf{z}_t' = (\mathbf{u}_t,\mathbf{z}_t)$ instead of simply $\mathbf{z}_t$.

Lemma 4. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log(2\pi_t/\delta) + 4d\log\left(dtbr\sqrt{\log 2da/\delta}\right)$, where $\sum_{t\geq 1} 1/\pi_t = 1$, $\pi_t > 0$. Let $\tau_t = dt^2br\sqrt{\log 2da/\delta}$, let $[\mathbf{s}_t^*]_t$ denote the closest point in S_t to $\mathbf{s}_t^*$. Hence,*

$$|f(\mathbf{s}_t^*,\mathbf{u}_t,\mathbf{z}_t) - \mu_{t-1}([\mathbf{s}_t^*]_t,\mathbf{u}_t,\mathbf{z}_t)| \leq \beta_t^{1/2}\sigma_{t-1}([\mathbf{s}_t^*]_t,\mathbf{u}_t,\mathbf{z}_t) + \frac{1}{t^2}, \tag{C.8}$$

for all $t \geq 1$, holds with probability at least $1-\delta$.

Lemma 5. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log(4\pi_t\delta) + 4d\log\left(dtbr\sqrt{4da/\delta}\right)$ where $\sum_{t\geq 1} 1/\pi_t = 1$, $\pi_t > 0$. Then, with probability at least $1-\delta$, for all $t \geq 1$, the multi-objective contextual is bounded as*

$$r_t \leq 2\beta_t^{1/2}\sigma_{t-1}(\mathbf{s}_t,\mathbf{u}_t,\mathbf{z}_t) + \frac{1}{t^2}. \tag{C.9}$$

To derive the final theorem's statements, as in the proof of Theorem 1 in (Krause & Ong, 2011), these lemmas are combined with Theorem 4 in (Krause & Ong, 2011) (i.e., also Theorem 1 in (Srinivas et al., 2009)). □

C.2 Additional plots

In this section, additional experimental results are included to complement those in the main text.

C.2.1 Preferences comparison

An experiment is run to show the impact of including the preferences in a single Gaussian Process (as our algorithm does), rather than having a separate Gaussian Process $f^{[u]}$ for each preference vector, as discussed in Remark 27 of the main text. The key difference here is that including the preferences in the kernel function for the Gaussian process allows learning about preference vectors together, rather than learning about each preference vector separately.

In Figure C.1 a comparison of the two approaches is established in terms of mean cumulative regret, averaged over 50 repetitions. One sees that learning using preferences as an input to our kernel in the Gaussian process is significantly faster than when considering a separate Gaussian process for each preference vector. In these experiments, the mean cumulative regret was 2.14 times larger after 40 iterations in the separate setting.

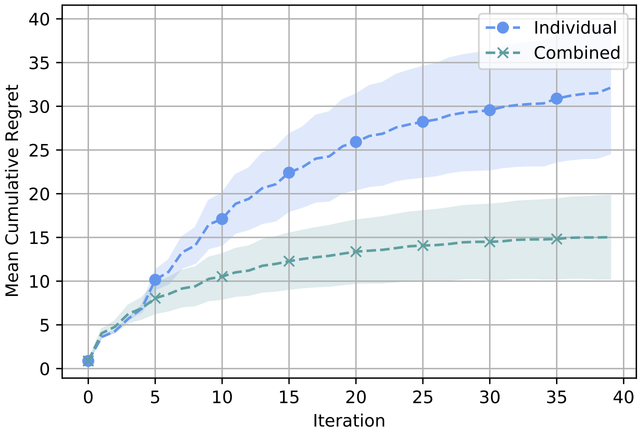

Figure C.1: Comparing CGP-UCB-MO with different approaches to preferences. The plot shows the mean cumulative regret in each case.

C.2.2 Kernel comparison

The difference in performance stemming from using different kernels in the Gaussian process is also measured. Figure C.2 shows the mean cumulative regret for different kernels, averaged over 60 repetitions. The RBF kernel performs the best, whilst the constant and linear kernels are the least suitable. This is usually task- and context-specific.

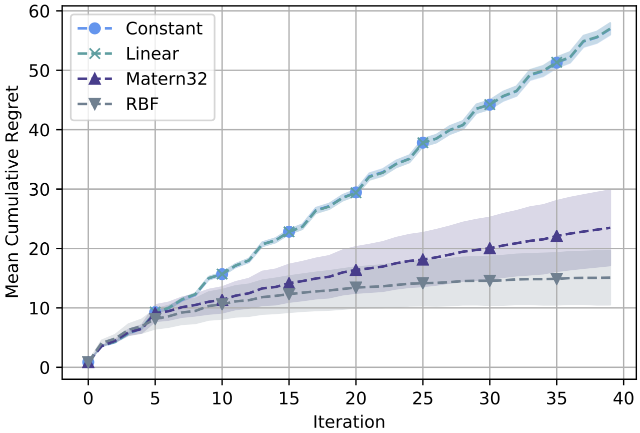

Figure C.2: Comparing CGP-UCB-MO with different kernel choices. The plot shows the mean cumulative regret in each case.

C.2.3 Mean average regret

Additional plots showing the mean average regret for each of the plots included in the main paper are provided, as well as the preference and kernel comparison plots. This is defined by taking the average of $\bar{R}_t$ over multiple runs. These plots are shown in Figures C.3 - C.6.

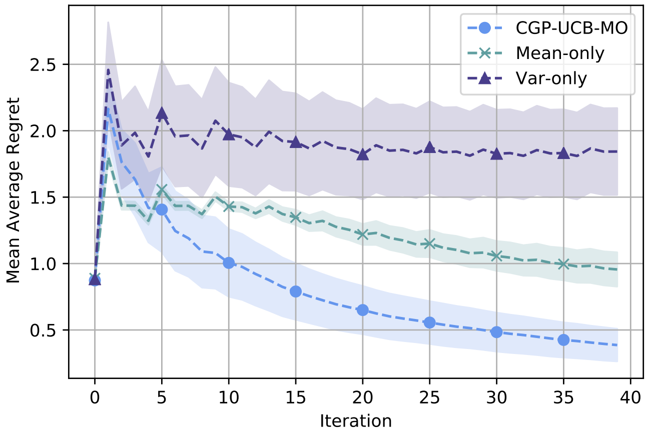

Figure C.3: Comparing CGP-UCB-MO with a mean or variance only approach when choosing the next action. The plot shows the mean average regret for each case in a single observation setting.

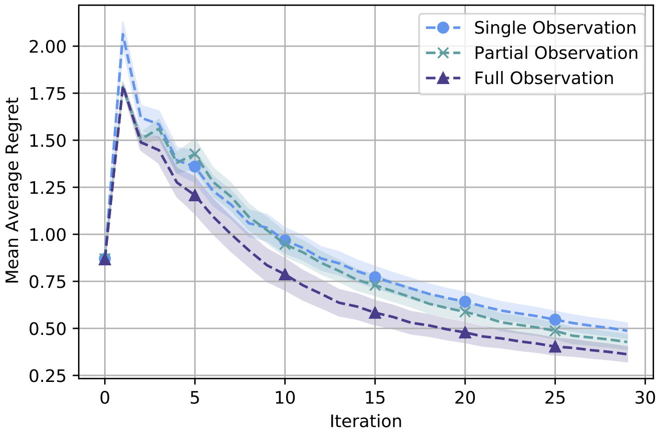

Figure C.4: Comparing CGP-UCB-MO with different levels of observability using coregionalisation. The plot shows the mean average regret for each case.

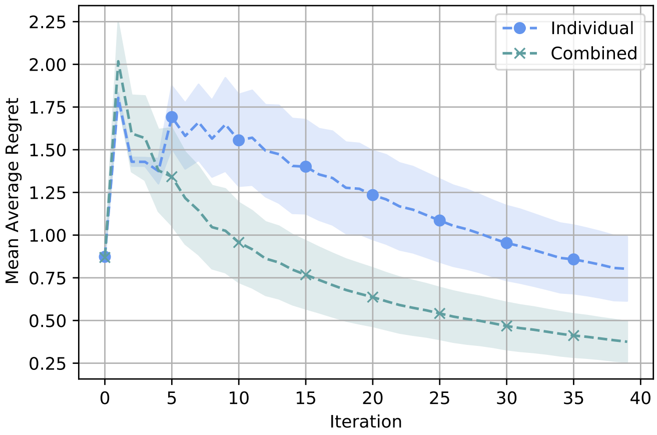

Figure C.5: Comparing CGP-UCB-MO with different approaches to preferences. The plot shows the mean average regret in each case.

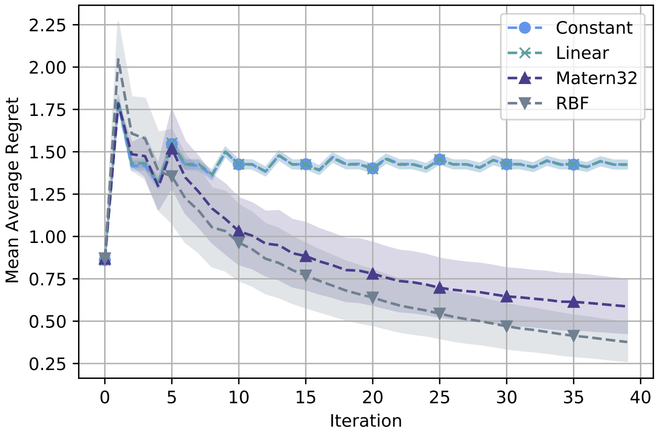

Figure C.6: Comparing CGP-UCB-MO with different kernel choices. The plot shows the mean average regret in each case.

Bibliography

Abbasi, M., Bhaskara, A., & Venkatasubramanian, S. Fair clustering via equitable group representations. *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, 504–514.

Abdolmaleki, A., Huang, S., Hasenclever, L., Neunert, M., Song, F., Zambelli, M., Martins, M., Heess, N., Hadsell, R., & Riedmiller, M. A distributional view on multi-objective policy optimization. *International conference on machine learning*. PMLR. 2020, 11–22.

Abels, A., Roijers, D., Lenaerts, T., Nowé, A., & Steckelmacher, D. Dynamic weights in multi-objective deep reinforcement learning. *International conference on machine learning*. PMLR. 2019, 11–20.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734–749.

Aggarwal, C. C. et al. (2016). *Recommender systems* (Vol. 1). Springer.

Agrawal, A., Pfisterer, F., Bischl, B., Chen, J., Sood, S., Shah, S., Buet-Golfouse, F., Mateen, B. A., & Vollmer, S. J. (2020). Debiasing classifiers: Is reality at variance with expectation? *Available at SSRN 3711681*.

Ahmadian, S., Epasto, A., Kumar, R., & Mahdian, M. Clustering without over-representation. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery data mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, 267–275.

Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, *3*, 5–40.

Anderson, T. W. (2004). *An introduction to multivariate statistical analysis* (Third Edition). Wiley.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., & Zaremba, W. (2017). Hindsight experience replay. *Advances in Neural Information Processing systems*, *30*.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*, 235–256.

Barocas, S., Hardt, M., & Narayanan, A. (2017). Fairness in machine learning. *Nips Tutorial*, *1*.

Barocas, S., & Selbst, A. (2016). Big data's disparate impact. *California Law Review*, *104*(1), 671–729.

Barrett, L., & Narayanan, S. Learning all optimal policies with multiple criteria. *Proceedings of the 25th international conference on machine learning*. 2008, 41–47.

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, *101*(473), 138–156.

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*(Nov), 463–482.

Basilico, J., & Hofmann, T. Unifying collaborative and content-based filtering. *Proceedings of the twenty-first international conference on machine learning*. 2004, 9.

Ben-Hamou, A., Boucheron, S., & Ohannessian, M. I. (2017). Concentration inequalities in the infinite urn scheme for occupancy counts and the missing mass, with applications. *Bernoulli*, *23*(1), 249 –287.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2018). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*.

Bernstein, D. S. Matrix mathematics. *Matrix mathematics*. Princeton University Press, 2009.

Bhagoji, A. N., Cullina, D., Sitawarin, C., & Mittal, P. Enhancing robustness of machine learning systems via data transformations. *2018 52nd annual conference on information sciences and systems (CISS)*. IEEE. 2018, 1–5.

Bingham, N. H., Goldie, C. M., Teugels, J. L., & Teugels, J. (1989). *Regular variation*. Cambridge University Press.

Biswas, S., & Rajan, H. (2020). Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness.

Blondel, M., Fujino, A., Ueda, N., & Ishihata, M. Higher-order factorization machines. *Proceedings of the 30th international conference on neural information processing systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, 3359–3367.

Bogunovic, I., Scarlett, J., & Cevher, V. Time-varying Gaussian process bandit optimization. *Artificial intelligence and statistics*. PMLR. 2016, 314–323.

Bonilla, E. V., Chai, K., & Williams, C. (2007). Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems*, *20*.

Boucheron, S., Bousquet, O., & Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: PS*, *9*, 323–375.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, *2*, 499–526.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brownlees, C., Joly, E., & Lugosi, G. (2015). Empirical risk minimization for heavy-tailed losses. *Ann. Statist.*, *43*(6), 2507–2536.

Buet-Golfouse, F. (2017). *Capital transactions* [London Graduate School in Mathematical Finance PhD Day]. https://www.londonmathfinance.org.uk/lgs-phd-day

Buet-Golfouse, F. Partially aware: Some challenges around uncertainty and ambiguity in fairness. *Advances in neural information processing systems. Workshop on fair AI in finance*. 2020. https://sites.google.com/view/faif2020/paper-download

Buet-Golfouse, F. ‘‘Art meets science”: Tackling data and perceptions. *KDD ’21: The 27th ACM SIGKDD conference on knowledge discovery and data mining. Workshop on understanding public perceptions for applied data science*. 2021. https://dl.acm.org/doi/10.1145/3447548.3469459

Buet-Golfouse, F. Asymmetry and heavy tails: Built-in robustness in classification. *International conference on learning representations. Robustml workshop*. 2021. https://sites.google.com/connect.hku.hk/robustml-2021/accepted-papers/paper-049

Buet-Golfouse, F. Narrow margins: Classification, margins and fat tails. *International conference on machine learning*. PMLR. 2021, 1127–1135.

Buet-Golfouse, F., & Hill, P. Optimal execution via multi-objective multi-armed bandits. *Proceedings of the AAAI conference on artificial intelligence*. *37*. 2023.

Buet-Golfouse, F., & Martin, N. W. Lifting Volterra diffusions via kernel decomposition. *Proceedings of the fourth acm international conference on AI in finance*. 2023, 481–489.

Buet-Golfouse, F., & Owen, A. (2016). The application of Hermite polynomials to risk allocation. *Journal of Risk*, *18*(3), 77 –110.

Buet-Golfouse, F., & Pahwa, P. Robust multi-objective reinforcement learning with dynamic preferences. *Asian conference on machine learning*. PMLR. 2023, 96–111.

Buet-Golfouse, F., & Roggeman, H. Numerical approximations of log Gaussian cox process. *Proceedings of the AAAI conference on artificial intelligence*. *36*. (11). 2022, 12923–12924.

Buet-Golfouse, F., Roggeman, H., & Utyagulov, I. Rayleigh portfolios and penalised matrix decomposition. *Companion proceedings of the web conference 2022*. 2022, 579–582.

Buet-Golfouse, F., Roggeman, H., & Utyagulov, I. Robust collaborative learning for sequence modelling. *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2022, 1146–1150.

Buet-Golfouse, F., & Utyagulov, I. Kernel factorisation machines. *2021 20th IEEE international conference on machine learning and applications (ICMLA)*. 2021, 1748–1753.

Buet-Golfouse, F., & Utyagulov, I. Towards fair multi-stakeholder recommender systems. *Adjunct proceedings of the 30th ACM conference on user modeling, adaptation and personalization*. 2022, 255–265.

Buet-Golfouse, F., & Utyagulov, I. Towards fair unsupervised learning. *2022 ACM conference on fairness, accountability, and transparency*. 2022, 1399–1409.

Buet-Golfouse, F., & Utyagulov, I. Towards fair unsupervised learning. *2022 ACM conference on fairness, accountability, and transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, 1399–1409.

Buet-Golfouse, F., & Utyagulov, I. Fairness trade-offs and partial debiasing. *Asian conference on machine learning*. PMLR. 2023, 112–136.

Buet-Golfouse, F., Utyagulov, I., Pahwa, P., & Hill, P. Turbo-charging deep learning methods for partial differential equations. *Proceedings of the fourth acm international conference on AI in finance*. 2023, 150–158.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*, 331–370.

Busa-Fekete, R., Szörényi, B., Weng, P., & Mannor, S. Multi-objective bandits: Optimizing the generalized Gini index. *International conference on machine learning*. PMLR. 2017, 625–634.

Calders, T., & Verwer, S. (2010). Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, *21*(2), 277–292.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R. Optimized pre-processing for discrimination prevention (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, Eds.). In *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, Eds.). Ed. by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. *30*. Long Beach, CA: Curran Associates, 2017, 3992–4001. http://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf

Cannelli, L., Nuti, G., Sala, M., & Szehr, O. (2020). Hedging using reinforcement learning: Contextual k-armed bandit versus q-learning. *arXiv preprint arXiv:2007.01623*.

Cartea, Á., Jaimungal, S., & Penalva, J. (2015). *Algorithmic and high-frequency trading*. Cambridge University Press.

Castelletti, A., Pianosi, F., & Restelli, M. Tree-based fitted q-iteration for multi-objective Markov decision problems. *The 2012 international joint conference on neural networks (IJCNN)*. IEEE. 2012, 1–8.

Celis, E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., & Vishnoi, N. Fair and diverse DPP-based data summarization (J. Dy & A. Krause, Eds.). In *Proceedings of the 35th international conference on machine learning* (J. Dy & A. Krause, Eds.). Ed. by Dy, J., & Krause, A. *80*. Proceedings of Machine Learning Research. PMLR, 2018, 716–725. https://proceedings.mlr.press/v80/celis18a.html

Chatterjee, S., & Hadi, A. S. (1986). Influential observations, high leverage points, and outliers in linear regression. *Statist. Sci.*, *1*(3), 379–393.

Chen, I. Y., Johansson, F. D., & Sontag, D. Why is my classifier discriminatory? *Proceedings of the 32nd international conference on neural information processing systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, 3543–3554.

Chen, J., Kallus, N., Mao, X., Svacha, G., & Udell, M. Fairness under unawareness: Assessing disparity when protected class is unobserved. *Proceedings of the conference on fairness, accountability, and transparency*. FAT* '19. Atlanta, GA, USA: Association for Computing Machinery, 2019, 339–348.

Chen, W., Wang, Y., & Yuan, Y. Combinatorial multi-armed bandit: General framework and applications. *International conference on machine learning*. PMLR. 2013, 151–159.

Chen, X., Ghadirzadeh, A., Björkman, M., & Jensfelt, P. Meta-learning for multi-objective reinforcement learning. *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2019, 977–983.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. Wide & deep learning for recommender systems. *Proceedings of the 1st workshop on deep learning for recommender systems*. DLRS 2016. Boston, MA, USA: Association for Computing Machinery, 2016, 7–10.

Cheng, W., Shen, Y., & Huang, L. Adaptive factorization network: Learning adaptive-order feature interactions. *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, new york, ny, usa, february 7-12, 2020*. AAAI Press, 2020, 3609–3616.

Chhabra, A., Masalkovaitė, K., & Mohapatra, P. (2021). An overview of fairness in clustering. *IEEE Access*, *9*, 130698–130720.

Chierichetti, F., Kumar, R., Lattanzi, S., & Vassilvitskii, S. Fair clustering through fairlets. *Proceedings of the 31st international conference on neural information processing systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 5036–5044.

Chouldechova, A. (2017a). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, *5*(2), 153–163.

Chouldechova, A. (2017b). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, *5*(2), 153–163.

Chuang, C.-Y., & Mroueh, Y. Fair mixup: Fairness via interpolation. *International conference on learning representations*. 2021. https://openreview.net/forum?id=DNl5s5BXeBn

Cooke, R. M., Nieboer, D., & Misiewicz, J. (2014). Regularly varying and subexponential distributions. *Fat-tailed distributions: Volume 1* (pp. 49–63). John Wiley & Sons, Ltd.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. Algorithmic decision making and the cost of fairness. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. KDD '17. Halifax, NS, Canada: Association for Computing Machinery, 2017, 797–806.

Crisafi, M. A., & Macrina, A. (2014). Optimal execution in lit and dark pools. *arXiv preprint arXiv:1405.2023*.

Dacrema, M. F., Boglio, S., Cremonesi, P., & Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM Trans. Inf. Syst.*, *39*(2).

Dai, S., Song, J., & Yue, Y. Multi-task Bayesian optimization via Gaussian process upper confidence bound. *ICML 2020 workshop on real world experiment design and active learning*. 2020. https://realworldml.github.io/files/cr/35_Camera_Ready_RealML.pdf

Damianou, A., & Lawrence, N. D. Deep Gaussian processes. *Artificial intelligence and statistics*. PMLR. 2013, 207–215.

DasGupta, A. (2008). *Asymptotic theory of statistics and probability* (Vol. 180). Springer.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance* (Vol. 1170). Springer.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J. S., & Pontil, M. Empirical risk minimization under fairness constraints (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, Eds.). In *Advances in neural information processing systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, Eds.). Ed. by Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. *31*. Curran Associates, Inc., 2018. https://proceedings.neurips.cc/paper/2018/file/83cdcec08fbf90370fcf53bdd56604ff-Paper.pdf

Drugan, M. M., & Nowe, A. Designing multi-objective multi-armed bandits algorithms: A study. *The 2013 international joint conference on neural networks (IJCNN)*. IEEE. 2013, 1–8.

Dua, D., & Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml

Duvenaud, D. (2014). *Automatic model construction with Gaussian processes* (Doctoral dissertation). University of Cambridge. https://www.repository.cam.ac.uk/handle/1810/247281

Duvenaud, D. K., Nickisch, H., & Rasmussen, C. (2011). Additive Gaussian processes. *Advances in Neural Information Processing Systems*, *24*.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. Fairness through awareness. *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, 214–226.

Efficient exploration in reinforcement learning. (1992). *Technical Report. Carnegie Mellon University*.

Ekstrand, M. D., Tian, M., Azpiazu, I. M., Ekstrand, J. D., Anuyah, O., McNeill, D., & Pera, M. S. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. *Conference on fairness, accountability and transparency*. PMLR. 2018, 172–186.

Fithian, W., & Mazumder, R. (2018). Flexible Low-Rank Statistical Modeling with Missing Data and Side Information. *Statistical Science*, *33*(2), 238 –260.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, *55*(1), 119–139.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. A comparative study of fairness-enhancing interventions in machine learning. *Proceedings of the conference on fairness, accountability, and transparency - FAT* '19*. New York, New York, USA: ACM Press, 2019, 329–338. arXiv: 1802.04422.

Friedman, E., & Fontaine, F. (2018). Generalizing across multi-objective reward functions in deep reinforcement learning. *arXiv preprint arXiv:1809.06364*.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407.

Fu, J., Co-Reyes, J., & Levine, S. (2017). Ex2: Exploration with exemplar models for deep reinforcement learning. *Advances in neural information processing systems*, *30*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of Machine Learning Research*, *17*(1), 2096–2030.

Ghadiri, M., Samadi, S., & Vempala, S. Socially fair k-means clustering. *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, 438–448.

Glasserman, P., & Xu, X. (2014). Robust risk measurement and model risk. *Quantitative Finance*, *14*(1), 29–58.

Glukhov, V. (2022). Reward is not enough: Can we liberate AI from the reinforcement learning paradigm? *arXiv preprint arXiv:2202.03192*.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT Press.

Gorski, J., Pfeuffer, F., & Klamroth, K. (2007). Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, *66*. https://doi.org/10.1007/s00186-007-0161-1

Guéant, O. (2016). *The financial mathematics of market liquidity: From optimal execution to market making* (Vol. 33). CRC Press.

Guiso, L., Sapienza, P., & Zingales, L. (2018). Time varying risk aversion. *Journal of Financial Economics*, *128*(3), 403–421.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. Deepfm: A factorization-machine based neural network for ctr prediction. *Proceedings of the 26th international joint conference on artificial intelligence*. IJCAI'17. Melbourne, Australia: AAAI Press, 2017, 1725–1731.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, *29*, 3323–3331.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (Second). Springer New York Inc.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015a). *Statistical learning with sparsity: The lasso and generalizations*. CRC press.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015b). *Statistical learning with sparsity: The lasso and generalizations*. Chapman Hall/CRC.

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., et al. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, *36*(26).

He, X., & Chua, T.-S. Neural factorization machines for sparse predictive analytics. *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: Association for Computing Machinery, 2017, 355–364.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. Neural collaborative filtering. *Proceedings of the 26th international conference on world wide web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, 173–182.

Hill, P., & Buet-Golfouse, F. Decomposing causality and fairness (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K., Liu, R., & Burns, T. F. OpenReview.net, 2023. https://openreview.net/pdf?id=Lm7z2vYergk

Hsu, D., & Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, *17*(18), 1–40. http://jmlr.org/papers/v17/14-273.html

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. *Workshop on faces in 'real-life' images: Detection, alignment, and recognition*. 2008.

Huber, P. J., & Ronchetti, E. M. (2009). *Robust statistics* (Second). Wiley.

Ibragimov, M., Ibragimov, R., & Walden, J. (2015). *Heavy-tailed distributions and robustness in economics and finance* (Vol. 214). Springer.

Imamura, H., Charoenphakdee, N., Futami, F., Sato, I., Honda, J., & Sugiyama, M. (2020). Time-varying Gaussian process bandit optimization with non-constant evaluation time. *arXiv preprint arXiv:2003.04691*.

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. Field-aware factorization machines for ctr prediction. *Proceedings of the 10th ACM conference on recommender systems*. RecSys '16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, 43–50.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, *33*(1), 1–33.

Kim, J. S., Chen, J., & Talwalkar, A. Model-agnostic characterization of fairness trade-offs. *Proceedings of the international conference on machine learning*. Vienna, Austria /

Online, 2020, June, 9339–9349. https://proceedings.icml.cc/paper/2020/hash/cf5530d9e441e0d78574353214373569

Kim, J. S., Chen, J., & Talwalkar, A. Model-agnostic characterization of fairness trade-offs. *Proceedings of the international conference on machine learning*. *37*. Online, 2020, 9339–9349.

Kleinberg, J., Mullainathan, S., & Raghavan, M. Inherent trade-offs in the fair determination of risk scores (C. H. Papadimitriou, Ed.). In *Proceedings of the 8th innovations in theoretical computer science conference* (C. H. Papadimitriou, Ed.). Ed. by Papadimitriou, C. H. *67*. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

Kleindessner, M., Awasthi, P., & Morgenstern, J. Fair k-center clustering for data summarization (K. Chaudhuri & R. Salakhutdinov, Eds.). In *Proceedings of the 36th international conference on machine learning* (K. Chaudhuri & R. Salakhutdinov, Eds.). Ed. by Chaudhuri, K., & Salakhutdinov, R. *97*. Proceedings of Machine Learning Research. PMLR, 2019, 3448–3457. https://proceedings.mlr.press/v97/kleindessner19a.html

Kleindessner, M., Awasthi, P., & Morgenstern, J. (2020). A notion of individual fairness for clustering. *arXiv preprint arXiv:2006.04960*.

Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J. Guarantees for spectral clustering with fairness constraints (K. Chaudhuri & R. Salakhutdinov, Eds.). In *Proceedings of the 36th international conference on machine learning* (K. Chaudhuri & R. Salakhutdinov, Eds.). Ed. by Chaudhuri, K., & Salakhutdinov, R. *97*. Proceedings of Machine Learning Research. PMLR, 2019, 3458–3467. https://proceedings.mlr.press/v97/kleindessner19b.html

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*(9), 992–1007.

Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*. 2008, 426–434.

Krause, A., & Ong, C. S. Contextual Gaussian process bandit optimization. *Proceedings of the 24th international conference on neural information processing systems*. 2011, 2447–2455.

Kuleshov, V., & Precup, D. (2014). Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*.

Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Li, K., Zhang, T., & Wang, R. (2020a). Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, *51*(6), 3103–3114.

Li, T., Sanjabi, M., Beirami, A., & Smith, V. Fair resource allocation in federated learning. *International conference on learning representations*. 2020. https://openreview.net/forum?id=ByexElSYDr

Li, Y., Chen, H., Fu, Z., Ge, Y., & Zhang, Y. User-oriented fairness in recommendation. *Proceedings of the web conference 2021*. WWW '21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, 624–632.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. Xdeepfm: Combining explicit and implicit feature interactions for recommender systems. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, 1754–1763.

Lin, J. G. (2005). On min-norm and min-max methods of multi-objective optimization. *Math. Program.*, *103*(1), 1–33.

Liu, C., Xu, X., & Hu, D. (2014). Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(3), 385–398.

Lugosi, G., & Mendelson, S. (2019). Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics*, *19*, 1145–1190.

Makarychev, Y., & Vakilian, A. Approximation algorithms for socially fair clustering. *Conference on learning theory*. PMLR. 2021, 3246–3264.

Mangasarian, O. L. (1999). Arbitrary-norm separating plane. *Operations Research Letters*, *24*(1-2), 15–23.

Martin, N. W. D., Hill, P., Tan, T. S., & Buet-Golfouse, F. Sustainable resource management (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K., Liu, R., & Burns, T. F. OpenReview.net, 2023. https://openreview.net/pdf?id=DLwlmWwmJBi

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering* (Vol. 38). M. Dekker New York.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, *54*(6), 1–35.

Melo, F. S. (2001). Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, 1–4.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *International conference on machine learning*. PMLR. 2016, 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, *abs/1312.5602*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. The MIT Press.

Mossalam, H., Assael, Y. M., Roijers, D. M., & Whiteson, S. (2016). Multi-objective deep reinforcement learning. *arXiv preprint arXiv:1610.02707*.

Narayanan, A. Translation tutorial: 21 fairness definitions and their politics. *Proceedings of the conference on fairness, accountability and transparency*. FAT* 18. New York, USA, 2018.

Natarajan, S., & Tadepalli, P. Dynamic preferences in multi-criteria reinforcement learning. *Proceedings of the 22nd international conference on machine learning*. 2005, 601–608.

Nielsen, F., & Sun, K. (2016). Guaranteed bounds on information-theoretic measures of univariate mixtures using piecewise log-sum-exp inequalities. *Entropy*, *18*(12).

Oneto, L., Donini, M., Pontil, M., & Maurer, A. Learning fair and transferable representations with theoretical guarantees. *2020 IEEE 7th international conference on data science and advanced analytics (dsaa)*. 2020, 30–39.

Oneto, L., & Chiappa, S. Fairness in machine learning. *Recent trends in learning from data: Tutorials from the inns big data and deep learning conference (innsbddl2019)*. Springer. 2020, 155–196.

Oneto, L., Donini, M., Pontil, M., & Shawe-Taylor, J. (2020). Randomized learning and generalization of fair and private classifiers: From pac-Bayes to stability and differential privacy. *Neurocomputing*, *416*, 231–243.

Pahwa, P., Thakur, K., & Buet-Golfouse, F. Dynamic human AI collaboration (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K., Liu, R., & Burns, T. F. OpenReview.net, 2023. https://openreview.net/pdf?id=Muwb2KohnX

Parisi, S., Pirotta, M., Smacchia, N., Bascetta, L., & Restelli, M. Policy gradient approaches for multi-objective sequential decision making. *2014 international joint conference on neural networks (IJCNN)*. IEEE. 2014, 2323–2330.

Pfisterer, F. (2022). *Democratizing machine learning: Contributions in automl and fairness* (Doctoral dissertation).

Pirotta, M., Parisi, S., & Restelli, M. Multi-objective reinforcement learning with continuous pareto frontier approximation. *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

Pirotta, M., Restelli, M., & Bascetta, L. (2015b). Policy gradient in Lipschitz Markov decision processes. *Machine Learning*, *100*(2), 255–283.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. On fairness and calibration (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, Eds.). In *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, Eds.). Ed. by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. *30*. Curran Associates, Inc., 2017, 5680–5689. http://papers.nips.cc/paper/7151-on-fairness-and-calibration.pdf

Polino, A., Pascanu, R., & Alistarh, D. (2018). Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*.

Qu, Y., Fang, B., Zhang, W., Tang, R., Niu, M., Guo, H., Yu, Y., & He, X. (2018). Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems*, *37*(1).

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. *Advances in neural information processing systems*, *20*.

Rasmussen, C. E. Gaussian processes in machine learning. *Summer school on machine learning*. Springer. 2003, 63–71.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning (adaptive computation and machine learning)*. The MIT Press.

Rawls, J. (1971). *A theory of justice* (1st ed.). Belknap Press of Harvard University Press.

Rendle, S. Factorization machines. *Proceedings of the 2010 IEEE international conference on data mining*. ICDM '10. USA: IEEE Computer Society, 2010, 995–1000.

Rendle, S., Krichene, W., Zhang, L., & Anderson, J. Neural collaborative filtering vs. matrix factorization revisited. *Fourteenth ACM conference on recommender systems*. RecSys '20. Virtual Event, Brazil: Association for Computing Machinery, 2020, 240–248.

Reymond, M., & Nowé, A. Pareto-dqn: Approximating the pareto front in complex multi-objective decision problems. *Proceedings of the adaptive and learning agents workshop (ala-19) at aamas*. 2019.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Wiele, T., Mnih, V., Heess, N., & Springenberg, J. T. Learning by playing solving sparse reward tasks from scratch. *International conference on machine learning*. PMLR. 2018, 4344–4353.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, *58*(5), 527–535.

Rodolfa, K. T., Salomon, E., Haynes, L., Mendieta, I. H., Larson, J., & Ghani, R. Case study: Predictive fairness to reduce misdemeanor recidivism through social service interventions. *Proceedings of the 2020 conference on fairness, accountability, and transparency*. FAT* '20. Barcelona, Spain: Association for Computing Machinery, 2020, January, 142–153.

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, *48*, 67–113.

Roijers, D. M., Whiteson, S., Vamplew, P., & Dazeley, R. Why multi-objective reinforcement learning. *European workshop on reinforcement learning*. 2015, 1–2.

Rosset, S., Zhu, J., & Hastie, T. Margin maximizing loss functions. *Proceedings of the 16th international conference on neural information processing systems*. NIPS'03. Whistler, British Columbia, Canada: MIT Press, 2003, 1237–1244.

Rosset, S., Zhu, J., & Hastie, T. (2004). Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, *5*, 941–973.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206–215.

Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, *71*(2), 319–392.

Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.

Samadi, S., Tantipongpipat, U., Morgenstern, J., Singh, M., & Vempala, S. The price of fair PCA: One extra dimension. *Proceedings of the 32nd international conference on neural information processing systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, 10999–11010.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on world wide web*. 2001, 285–295.

Schafer, J. B., Konstan, J., & Riedl, J. Recommender systems in e-commerce. *Proceedings of the 1st acm conference on electronic commerce*. 1999, 158–166.

Schapire, R. E., Freund, Y., Barlett, P., & Lee, W. S. Boosting the margin: A new explanation for the effectiveness of voting methods. *Proceedings of the fourteenth international conference on machine learning*. ICML '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, 322–330.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*. 2002, 253–260.

Scholkopf, B., & Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Seneta, E. (2002). Karamata's characterization theorem, Feller, and regular variation in probability theory. *Publications de l'institut mathematique*, *71*(85), 79–89.

Shah, A., Wilson, A., & Ghahramani, Z. Student-t processes as alternatives to Gaussian processes (S. Kaski & J. Corander, Eds.). In *Proceedings of the seventeenth international conference on artificial intelligence and statistics* (S. Kaski & J. Corander, Eds.). Ed. by Kaski, S., & Corander, J. *33*. Proceedings of Machine Learning Research. Reykjavik, Iceland: PMLR, 2014, 877–885. http://proceedings.mlr.press/v33/shah14.html

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shen, B., Gnanasambandam, R., Wang, R., & Kong, Z. J. (2022). Multi-task Gaussian process upper confidence bound for hyperparameter tuning and its application for simulation studies of additive manufacturing. *IISE Transactions*, 1–13.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, *529*(7587), 484–489.

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, *299*, 103535.

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., & Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. *Proceedings of the 28th ACM international conference on information and knowledge management*. CIKM '19. Beijing, China: Association for Computing Machinery, 2019, 1161–1170.

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Sutton, R. S. (1995). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, *8*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task Bayesian optimization. *Advances in neural information processing systems*, *26*.

Taleb, N. N. (2020). Statistical consequences of fat tails: Real world preasymptotics, epistemology, and applications. https://arxiv.org/pdf/2001.10488.pdf

Tantipongpipat, U., Samadi, S., Singh, M., Morgenstern, J. H., & Vempala, S. Multi-criteria dimensionality reduction with applications to fairness (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett, Eds.). In *Advances in neural information processing systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett, Eds.). Ed. by Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., & Garnett, R. *32*. Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper/2019/file/2201611d7a08ffda97e3e8c6b667a1bc-Paper.pdf

Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., & Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, *30*.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3-4), 285–294.

Udell, M., Horn, C., Zadeh, R., & Boyd, S. (2016). Generalized low rank models. *Foundations and Trends in Machine Learning*, *9*(1).

US Congress. (2003). P. L. 108-159: Fair and Accurate Credit Transactions Act. https://www.gpo.gov/fdsys/pkg/PLAW-108publ159/pdf/PLAW-108publ159.pdf

Utyagulov, I., Buet-Golfouse, F., & Hill, P. Fairness under partial observability (K. Maughan, R. Liu, & T. F. Burns, Eds.). In *The first tiny papers track at ICLR 2023, tiny papers @ ICLR 2023, Kigali, Rwanda, may 5, 2023* (K. Maughan, R. Liu, & T. F. Burns, Eds.). Ed. by Maughan, K., Liu, R., & Burns, T. F. OpenReview.net, 2023. https://openreview.net/pdf?id=if1Mmrxf-pq

Vakili, S., Moss, H., Artemev, A., Dutordoir, V., & Picheny, V. (2021). Scalable Thompson sampling using sparse Gaussian process models. *Advances in Neural Information Processing Systems*, *34*, 5631–5643.

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., & Dekker, E. (2011). Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, *84*, 51–80.

Vamplew, P., Smith, B. J., Källström, J., Ramos, G., Rădulescu, R., Roijers, D. M., Hayes, C. F., Heintz, F., Mannion, P., Libin, P. J., et al. (2022). Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, *36*(2), 41.

van Seijen, H., Fatemi, M., Romoff, J., Laroche, R., Barnes, T., & Tsang, J. Hybrid reward architecture for reinforcement learning. *Proceedings of the 31st international conference on neural information processing systems*. 2017, 5398–5408.

Van Meteren, R., & Van Someren, M. Using content-based filtering for recommendation. *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*. *30*. Barcelona. 2000, 47–56.

Van Moffaert, K., Drugan, M. M., & Nowé, A. Scalarized multi-objective reinforcement learning: Novel design techniques. *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (adprl)*. IEEE. 2013, 191–199.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Verma, S., & Rubin, J. Fairness definitions explained. *Proceedings of the international conference on software engineering*. New York, NY, USA: ACM, 2018, 1–7.

Wang, H., Li, G., & Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, *25*(3), 347–355.

Wang, L., & Wang. (2021). Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1995–2008.

Wang, R., Fu, B., Fu, G., & Wang, M. Deep cross network for ad click predictions. *Proceedings of the AdKDD'17*. AdKDD'17. Halifax, NS, Canada: Association for Computing Machinery, 2017.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MIT Press Cambridge, MA.

Wulfmeier, M., Abdolmaleki, A., Hafner, R., Springenberg, J. T., Neunert, M., Hertweck, T., Lampe, T., Siegel, N., Heess, N., & Riedmiller, M. (2019). Compositional transfer in hierarchical reinforcement learning. *arXiv preprint arXiv:1906.11228*.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T.-S. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *Proceedings of the 26th international joint conference on artificial intelligence*. IJCAI'17. Melbourne, Australia: AAAI Press, 2017, 3119–3125.

Xu, H., & Mannor, S. (2012). Robustness and generalization. *Machine learning*, *86*, 391–423.

Yahyaa, S. Q., & Manderick, B. Thompson sampling for multi-objective multi-armed bandits problem. *European symposium on artificial neural networks, computational intelligence and machine learning (ESANN)*. 2015, 47–52.

Yang, R., Sun, X., & Narasimhan, K. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Proceedings of the 33rd international conference on neural information processing systems*. 2019, 14636–14647.

Yeh, I.-C., & hui Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, *2*(36), 2473–2480.

Zafar, M. B., Valera, I., Rogriguez, M. G., & Gummadi, K. P. Fairness constraints: Mechanisms for fair classification (A. Singh & J. Zhu, Eds.). Ed. by Singh, A., & Zhu, J. *54*. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2017, 962–970.

Zafar, M. B., Valera, I., Rogriguez, M. G., & Gummadi, K. P. Fairness constraints: Mechanisms for fair classification (A. Singh & J. Zhu, Eds.). In *Proceedings of the 20th international conference on artificial intelligence and statistics* (A. Singh & J. Zhu, Eds.). Ed. by Singh, A., & Zhu, J. *54*. Proceedings of Machine Learning Research. PMLR, 2017, 962–970. https://proceedings.mlr.press/v54/zafar17a.html

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. Learning fair representations. *Proceedings of machine learning research*. *28*. (2). 2013, 1362–1370. http://proceedings.mlr.press/v28/zemel13.html

Zhacheny. (2019). Optimization based on Gini index for multi-objective bandits. https://github.com/zhacheny/Optimization-based-on-GNI-Index-For-multi-objective-bandits

Zhang, F. (2006). *The Schur complement and its applications* (Vol. 4). Springer Science & Business Media.

Zhang, K., Tsang, I. W., & Kwok, J. T. Improved Nyström low-rank approximation and error analysis. *Proceedings of the 25th international conference on machine learning*. 2008, 1232–1239.

Zhang, L., Shen, W., Huang, J., Li, S., & Pan, G. (2019). Field-aware neural factorization machine for click-through rate prediction. *IEEE Access*, *7*, 75032–75040.

Zhang, W., Du, T., & Wang, J. Deep learning over multi-field categorical data (N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, & G. Silvello, Eds.). In *Advances in information retrieval* (N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, & G. Silvello, Eds.). Ed. by Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., & Silvello, G. Cham: Springer International Publishing, 2016, 45–57.

Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586–5609.

Zhu, Z., Wang, J., & Caverlee, J. Measuring and mitigating item under-recommendation bias in personalized ranking systems. *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. SIGIR '20. Virtual Event, China: Association for Computing Machinery, 2020, 449–458.

Žliobaitė, I. (2017). Measuring discrimination in algorithmic decision making. *Data Mining and Knowledge Discovery*, *31*(4), 1060–1089.

Zuluaga, M., Krause, A., & Püschel, M. (2016). ε-pal: An active learning approach to the multi-objective optimization problem. *The Journal of Machine Learning Research*, *17*(1), 3619–3650.