Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

On the fairness-utility trade-off in consequential decision making under imperfect data

Floyd Kretschmar
Matrikelnummer: 4205979

July 21, 2020

Bearbeitungszeitraum: 01.03.2020 - 01.09.2020

Erstgutachter: Prof. Dr. Philipp Hennig, Universität Tübingen
Zweitgutachter: Prof. Dr. Isabel Valera, Universität des Saarlandes

ERKLÄRUNG

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Datum, Ort, Unterschrift

Contents

1 Summary and Contributions

The field of machine learning is and has always been a vast and diverse field of methods to allow machines to extract patterns of behaviour from data, and apply them in a wide array of tasks. Ever since the authors of Krizhevsky et al. (2012) showed the real-world applicability of deep neural-networks, one subset of the large corpus of machine learning methods, the field has seen a large increase in attention. This increased attention has in turn lead to an increased amount time and money spent on research which has yielded many novel or improved approaches on how to apply machine learning to a large number of real world applications. These methods aim to both aid or even fully replace humans in many day-to-day tasks. Given this development a growing number of people are asking questions about the societal implications of the potentially large scale changes brought about by these advancements in technology. These challenges range from discussions about the meaning of privacy in an age of machine learning systems trained on big data to explorations of the question of how to make decisions made by machine learning system explainable to humans.

One of the areas that has received a lot of attention in recent years concerns the question of how to enforce fairness in machine learning systems. Or more specifically how to prevent discrimination by machine learning systems. One subset of real-world applications this area is called "consequential decision making" which describes decision making tasks, that have long term effects on the people affected by the decisions. Such tasks range from pre-trial release decisions made by judges using risk scores generated by predictive models, over loan-decisions made by banks based on default probability to insurance companies basing costs of insurance policies based on predictive risk assessment. A growing number of works examine the impact that machine learning systems have on perpetuating discrimination in these kind of decision making tasks. The fundamentals of both fair machine learning in general as well as an introduction to consequential decision making can be found in chapter 2 of this thesis. The main publication that the work in this thesis is based upon is the recent paper by Kilbertus et al. (2019), which investigates the question of how to enforce fairness in consequential decision making, given imperfect data. In this context imperfect data means, that the data collection to train a given machine learning system was biased by some initial data collection policy. The authors argue, that in order to make truly fair decisions, that are also optimal in terms of the utility they provide to the decision maker, the paradigm has shift from "learning to predict" to "learning to decide". They introduce a framework in which they train decision making policies based on an optimization target that combines both utility and fairness considerations. Chapter 3 introduces their frame-

work, and evaluates its performance by applying it to a range of practical consequential decision making applications.

The main goal of this thesis then is to extend this framework to answer the following question: "Can one find an automatic, principled way to chose the trade-off (constant) between utility of the decision maker and fairness considerations?". To answer this question chapter 4 will explore both the theoretical as well as the practical implications of this question. The main contributions of the thesis meant to answer this question are as follows:

- Introduce the Lagrangian based Dual Gradient algorithm as well as its extension the Augmented Lagrangian Method to learn the trade-off constant between fairness and utility proposed by Kilbertus et al. (2019).
- Replace the original fairness constraint proposed by Kilbertus et al. (2019) with a relaxed version, inspired by the work of Zafar et al. (2019).
- Evaluate the performance both of the original framework, as well as the proposed extensions on a variety of different datasets.
- Provide a publicly available Python implementation of all methods presented in this thesis.

2 Introduction to Fairness in ML

Recognizing and exploiting patterns in data is at the very core of machine learning. If the data from which these patterns are extracted contain discriminatory or biased patterns, this will be reflected in the resulting algorithms. One of the goals of fair machine learning research is to find ways of quantifying and operationalizing fairness in a way that makes these biases obvious and allows to account for them. This becomes especially important, as consequential decision making tasks, "where decisions have significant consequences for individuals"Kilbertus et al. (2019), are being partially or fully automated by data-driven machine learning models. As these kind of tasks have substantial impact on the lives of the individuals for which decisions are made, a heightened focus is rightfully placed on ensuring that the underlying predictive models are not unfair in the sense that they do not to systematically disadvantage certain demographic groups based on shared sensitive characteristics, such as age, gender or race.

This necessity has resulted in a growing number of publications over the last couple of years that have started to explore how machine learning systems have been unknowingly perpetuating biased decisions making processes of the past. Two of the potentially most pressing questions, that these works have tried two answer are:

1. What (human) notion of fairness is most applicable to a given scenario, in which machine learning is used?
2. How can these notions of fairness be codified in a way, such that they can be enforced by a machine learning algorithm?

Especially the first of these questions, is a more existential one, that cannot be solved by researchers and scientists alone. Right now, a lot of the work and discussion has been centered around a notion of fairness, that tries to enforce non-discrimination. But the discussion surrounding whether or not that is the best way of defining fairness is still ongoing and it is lead by stakeholders from all kinds of disciplines such as philosophy, ethics, economics and politics. Machine learning researches are certainly playing their part in this discussion as well, but more so than the first question, research in the area of fair machine learning has been focused on the second question: how to codify and enforce the notions of fairness, agreed upon by society, in such a way that they can be measured and enforced within machine learning systems?

2.1 Operationalizing Fairness

As discussed before, the question of how to operationalize fairness, does depend on the notion of fairness chosen, but it also depends on the setting in which fairness is supposed to be enforced. In the previous section, the idea of consequential decision making has already been introduced informally. Formally these kinds of applications can be seen as special cases of binary classification described in Kilbertus et al. (2019): Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature domain, $\mathcal{S} = \{0, 1\}$ the range of sensitive attributes and $\mathcal{Y} = \{0, 1\}$ the set of ground truth labels. For example, a non-sensitive attribute could be number of prior convictions in the case of criminal risk assessment or the yearly income of an individual when making loan-decisions. It is assumed that for each individual $\boldsymbol{x} \in \mathcal{X}$, $s \in \mathcal{S}$ and $y \in \mathcal{Y}$ are given by the joint probability distribution $P(\boldsymbol{x}, s, y) = P(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)$. Within this framework a consequential decision $d \in \{0, 1\}$ is defined as the outcome of a machine learning system $Q(\hat{y} \mid x, s)$ for a given individual defined by its non-sensitive attributes $\boldsymbol{x}$ and sensitive attribute s.

A multitude of frameworks to codify fairness within this setting have been developed, the most prominent of which are arguably:

- **Individual Fairness:** Similar individuals, should be treated similarly, as proposed by Dwork et al. (2011)
- **Counterfactual Fairness:** Counterfactual explanations should have similar cost across different subgroups, as proposed by Kusner et al. (2017)
- **Group Fairness:** The outcomes of a decision making system should not systematically differ between social salient groups

This thesis will focus on group fairness as the fundamental paradigm that will be explored. What follows is an introduction on how group fairness can be used, to operationalize the notion of non-discrimination across subgroups of a population, using the example of criminal justice risk assessment as explained by Berk et al. (2018). The fairness definitions under group fairness are based on the accuracy measurements defined by the following confusion matrix:

$N = t_p + f_p + t_n + f_n$ Sample Size	$\hat{Y} = 1$ Positive Prediction	$\hat{Y} = 0$ Negative Prediction	Conditional Procedure Accuracy
$Y = 1$ Positive sample	t_p True Positive	f_n False Negative	$\frac{t_p}{t_p+f_n}$ True Positive Rate
$Y = 0$ Negative sample	f_p False Positive	t_n True Negative	$\frac{t_n}{t_n+f_p}$ True Negative Rate
Conditional Use Accuracy	$\frac{t_p}{t_p+f_p}$ Positive Predictive Value	$\frac{t_n}{t_n+f_n}$ Negative Predictive Value	$\frac{t_p+t_n}{t_p+f_p+t_n+f_n}$ Overall Accuracy

More specifically, Berk et al. (2018) define multiple measures of group fairness as the enforcement of equality of different accuracy measures across all protected group categories. Imposing such group equality constraints for different measures defined by the confusion matrix, leads to the following definitions of fairness:

1. **Overall accuracy equality:** equal probability of correct classification $\frac{t_p+t_n}{N}$ across subgroups.
2. **Statistical parity:** equal probability of positive or negative prediction $\frac{t_p+f_p}{N}$ or $\frac{t_n+f_n}{N}$ across subgroups. This is also known as **Demographic parity** and can be expressed in terms of the setting defined in the beginning of this section as $Q(\hat{y} = 1 \mid s = 0) = Q(\hat{y} = 1 \mid s = 1)$.
3. **Conditional procedure accuracy equality:** equal probability of correct classification, given the actual outcomes $\frac{t_p}{t_p+f_n}$ or $\frac{t_n}{t_n+f_p}$ across subgroups. Enforcing both of these conditions at the same time is known as Equalized Odds first introduced by Hardt et al. (2016), which can be relaxed to the notion of **Equality of Opportunity** where only equal probability of positive classification, given an actual positive outcome is enforced. This can be expressed in terms of the binary classification setting as $Q(\hat{y} = 1 \mid y = 1, s = 0) = Q(\hat{y} = 1 \mid y = 1, s = 1)$
4. **Conditional use accuracy equality:** equal probability of an outcome, given the prediction $\frac{t_p}{t_p+f_p}$ or $\frac{t_n}{t_n+f_n}$ across subgroups
5. **Treatment equality:** equal ratio between false positives and negatives $\frac{f_p}{f_n}$ or $\frac{f_n}{f_p}$ across subgroups
6. **Total fairness:** all previously defined notions of group fairness achieved simultaneously

Given the fact, that all of these definitions of fairness restrict the space of possible solutions of the original optimization problem in non-trivial ways, the question arises how they affect each other and what trade-offs between accuracy and fairness as well as between different kind of fairness are necessary. Berk et al. (2018) state that "[...] excluding S will reduce accuracy. Any procedure that even just discounts the role of S will lead to less accuracy.". They also explore the conflict between conditional use and procedure accuracy equality by citing the following impossibility theorem: "When the base rates[1] differ by protected group and when there is not separation[2], one cannot have both conditional use accuracy and equality in the false negative and false positive rates." They suggest, that "the key trade-off will be between the false positive and false negative rates on the one hand and the conditional use accuracy on the other." Equal results can be proven for the interaction between other group fairness measures as well, like shown by Barocas et al. (2019).

[1] proportion of actual failures/successes $\frac{t_p+f_n}{N}$ or $\frac{t_n+f_p}{N}$

[2] separation meaning that "perfectly accurate classification is possible" Berk et al. (2018)

Finally, the procedure with which any of these group fairness measures is actually enforced in a machine learning system, usually falls into one of the following three main strategies laid out by Berk et al. (2018):

1. **Pre-Processing** is the elimination of sources of unfairness in the data before formulating Q.
2. **In-Processing** means including the adjustments for fairness in the process of constructing Q.
3. In **Post-Processing** a potentially unfair Q is applied first, and its results are adjusted afterwards to account for fairness.

Each of these approaches comes with its own advantages and disadvantages, but the strategies explored by this thesis, are all part of the domain of in-processing.

2.2 Applications and Datasets

As touched on previously, consequential decision making encompasses a large variety of tasks, which have in common that a (binary) decision with potentially wide ranging consequences for the affected individual has to be made. This section is going to introduce some of the real world applications that fall under the umbrella term of consequential decision and it will also introduce some well known datasets associated with these tasks. In the course of this thesis these datasets will be used to evaluate the performance of the introduced methods.

2.2.1 Credit and lending

One of the quintessential examples of consequential decision making is decision making about loan applicants. In many cases the decision maker like a bank bases the loan decisions on credit scores. These have often been generated by some proprietary system or process, which means the full training data and/or functional form of the score might not be available to the decision maker. There might be a multitude of reasons, why the access to this information might be limited such as privacy or intellectual property concerns.

One example of such a scenario is the TransRisk credit score that has been first described by Hardt et al. (2016). It is a credit score by TransUnion, which is generated by a proprietary third party model created by the company FICO. It is similar to the SCHUFA score and other credit risk scores, in the sense that it a singular number, that can be used by decision makers, in their decision making processes. The FICO score was first used by Hardt et al. (2016) in their paper introducing equalized odds and equality of opportunity and then further expanded on as an example in Barocas et al. (2019). They analyse aggregate data published by the U.S. Federal Reserve in U.S. Federal Reserve (2007) about the relationship between the credit score, demographic

information such as race or gender and the performance of an individual given the following performance criterion:

> "(the) measure is based on the performance of new or existing accounts and measures whether individuals have been late 90 days or more on one or more of their accounts or had a public record item or a new collection agency account during the performance period."U.S. Federal Reserve (2007)

The group wise receiver operating characteristic (ROC) curves as seen in figure 2.1 can give an indication regarding the fairness of such a proprietary score. The ROC curve plots the false positive rate on the x-axis against the true positive rate on the y-axis for a range of different decision thresholds on the score function. That means one particular point on any of the ROC curves plotted in figure 2.1 represents the predictive performance of the score with regards to TPR and FPR for a given threshold. A perfect predictor would lie in the top left corner, meaning a true positive rate of 1 without any false positives. ROC curves therefore give a rough comparative estimate of the predictive performance of the FICO score across groups, as the ROC curve for white individuals is for example significantly closer to the perfect predictor, than for black individuals. This implies, that the score is unfair in the sense, that it is worse at predicting the performance of some demographic groups than others.

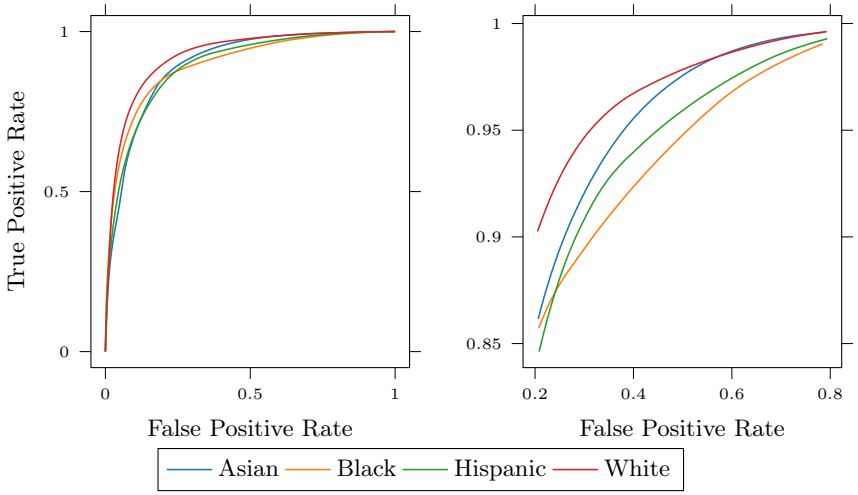

Figure 2.1: ROC Curves of the FICO dataset

Additional data is not provided on a per individual basis but instead as aggregated statistics in form of cumulative density functions across demographic groups. Therefore a new datasets can easily be generated by drawing samples from the known underlying distributions.

2.2.2 Criminal risk assessment

The second application that falls under the umbrella term of consequential decision making is criminal risk assessment and the decisions based on these assessments, such as bail or early release decisions. In the U.S. many of the decisions made within the criminal justice process are already being assisted or fully automated by criminal risk assessment systems, like the COMPAS risk assessment software. Similar to the FICO score discussed previously, the COMPAS risk assessment algorithm takes a certain set of sensitive attributes like the sex and the race of of an individual as well as some non-sensitive attributes like the number of felony and misdemeanor convictions they have accrued, and generates a risk score, that can be used as part of the decision making process by a judge or other human decision makers.

In Julia Angwin & Kirchner (2016) the authors analyzed 6172 samples of risk scores they had gained access to and compared the results to the true recidivism data of the given individuals. Of the attributes specified in the dataset this work will focus on the sensitive attribute race with the protected group being people of color and the unprotected group white individuals. The non-sensitive attributes used are juvenile offenses (number of felonies, misdemeanors and others), the number of prior convictions overall as well as the degree of the charge for which the risk assessment was created.

2.2.3 Advertisement strategy based on income

Advertisement delivery constitutes another scenario where machine learning systems are being used to automate decision making processes. Companies might want to tailor their advertisement strategy based on how affluent their target demographic is, as to target the right audience given their product or service portfolio. These kind of companies might have access to large databases of potential customers that include both non-sensitive attributes such as occupation or marital status as well as sensitive attribute such as gender or race. Based on this information they might devise a machine learning system that decides whether or not a certain individual is part of their desired target audience and is therefore advertised to.

One dataset with which this kind of scenario can be simulated is the Adult dataset made available by Blake & Merz (1998). This dataset is comprised of 48842 samples of a large number of non-sensitive and sensitive attributes. For the purpose of this thesis the sensitive attribute used is race where the protected group are people of color, and the unprotected group are white individuals. The non-sensitive attributes used are capital gains and losses, number of work-hours per week, type of work, education, marital status and country of birth.

3 Fair decision making

In the previous chapter consequential decision making as a field of application of machine learning has been introduced both in abstract terms and by introducing real world application, in which machine learning systems are already used today, to aid human decision making. Additionally the importance of fairness considerations in these kinds of applications have been established. This naturally gives rise to the question of how to design and train such machine learning systems.

Lets recall the consequential decision scenario as defined in 2.1 where the goal is to make a decision $d \in \{0, 1\}$ for an individual with both non-sensitive attributes $\boldsymbol{x} \in \mathcal{X}$ and sensitive attributes $s \in \mathcal{S}$. To achieve that goal a decision making policy $\pi(d \mid \boldsymbol{x}, s)$ has to be defined, from which decisions d can be sampled. It is assumed that the true performance $y \in \mathcal{Y}$ as well as $\boldsymbol{x}$ and s are given by the joint probability distribution $P(\boldsymbol{x}, s, y) = P(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)$. In this scenario the goal is to maximize the utility for the decision maker while also enforcing fairness according to some chosen fairness constraint. Corbett-Davies et al. (2017) propose the following function to measure what they call immediate utility

$$\begin{aligned} u_P(\pi) &= \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},s,y),d\sim\pi(d|\boldsymbol{x},s)}[yd - cd] \\ &= \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},s,y),d\sim\pi(d|\boldsymbol{x},s)}[d(y - c)] \\ &= \mathbb{E}_{\boldsymbol{x},s\sim P(\boldsymbol{x},s)}[\pi(d = 1 \mid \boldsymbol{x}, s)(P(y = 1 \mid \boldsymbol{x}, s) - c)] \\ &= \int (\pi(d = 1 \mid \boldsymbol{x}, s)(P(y = 1 \mid \boldsymbol{x}, s) - c))P(\boldsymbol{x}, s)\, dx\, ds \end{aligned} \tag{3.1}$$

where $c \in (0, 1)$ represents the economic cost of a (wrong) decision by the decision maker. They call this immediate utility as it "reflects only the proximate costs and benefits of decisions. It does not, for example, consider the long-term, systemic effects of a decision rule." Enforcing one the fairness criteria introduced in section 2.1 while maximizing the utility for the decision maker gives rise to the following constrained optimization problem

$$\begin{aligned} \underset{\theta}{\text{argmax}} \quad & u_P(\pi) \\ s.t. \quad & \mathcal{F}_P(\pi) \geq \delta \end{aligned} \tag{3.2}$$

where $\mathcal{F}$ is a function measuring unfairness according to the chosen fairness criterion and δ is a trade-off constant that defines the amount of unfairness that is to be allowed

within the decision making system. This fairness function can be chosen in a multitude of ways and this work, especially chapter 4, will explore different formulations of $\mathcal{F}$, their mathematical properties with regards to optimization as well as their comparative performance. This general problem of fair decision making can be approached in multiple ways, two of which will be introduced in the following sections.

3.1 Fair decision making as prediction

One of the most prolific approaches when using machine learning for consequential decision making is to split the problem it into two separate tasks:

1. **Predicting:** Predict the performance $\hat{Y}$ of individuals with a trained predictive model Q_θ
2. **Deciding:** Make decisions $d \in \{0, 1\}$ based in the predictions $\hat{Y}$ according to some decision policy π

Within this setting a **prediction task** is defined as predicting the label $y \in \mathcal{Y}$ for a given individual given its non-sensitive attributes $\boldsymbol{x} \in \mathcal{X}$ as well as its sensitive attributes $s \in \mathcal{S}$ for which the true label is unknown. To achieve that, a predictive model $Q(y \mid x, s; \theta)$ is trained in a supervised manner, which means fitting the parameters θ of the model by solving the optimization problem

$$\begin{aligned} \underset{\theta}{\text{argmin}} \quad & \mathcal{L}(Q_\theta) \\ s.t. \quad & \mathcal{F}(\pi_Q) \geq \alpha \end{aligned} \tag{3.3}$$

where $\mathcal{L}$ is the loss function of the predictive model $Q_\theta(y \mid x, s)$. Q_θ which is trained to be an approximation of the true underlying probability distribution $P(\boldsymbol{x}, s, y)$, in the sense that $Q_\theta(y = 1 \mid \boldsymbol{x}, s) \approx P(y = 1 \mid \boldsymbol{x}, s) - \delta_s$. This means Q_θ approximates the probability distribution defining the probability of an individual being a member of a specified class y. In the real world example given before of a bank wanting to make a decision on whether or not an individual is given a loan or not, the predictive model could for example assign an individual one of the classes $y = 0 =$ "will default on loan" or $y = 1 =$ "will not default on loan" with a certain probability.

The second part of this framework is about **making a decisions** using a decision making policy π and the predictions made by the predictive model $Q_\theta(y \mid x, s)$. Kilbertus et al. (2019) define such a policy as a mapping $\pi : \mathcal{X} \times \mathcal{S} \to \mathcal{P}(0, 1)$ that maps an individuals features to a probability distribution over binary decisions $d \in 0, 1$. The simplest choice for such a decision policy is a deterministic threshold policy of the form

$$\pi_Q(d = 1 \mid \boldsymbol{x}, s) = \mathbf{1}[Q(y = 1 \mid \boldsymbol{x}, s) \geq c] \tag{3.4}$$

where the policy π_Q makes a positive decision $d = 1$ for all individuals for which the trained model Q predicts the desirable performance class $\hat{Y} = 1$ with a confidence, that exceeds a certain confidence threshold c. This approach directly incorporates fairness constraints by allowing group specific δ_s and therefore indirectly allowing for different confidence thresholds c of the predictive model for different groups. For the loan example that means all individuals for which the predictive model has a confidence greater than c that the individual is part of the class "will not default on loan", a loan is given out by the bank.

According to Corbett-Davies et al. (2017) given perfect knowledge of the true ground truth probability distribution $P(y \mid \boldsymbol{x}, s)$ the optimal policy $\pi*$ solving 3.2 is a threshold policy as defined by 3.4. But in many real-world applications, the true ground truth distribution $P(y \mid \boldsymbol{x}, s)$ is not known and $Q(y = 1 \mid \boldsymbol{x}, s) \approx P(y = 1 \mid \boldsymbol{x}, s) - \delta_s$ is only approximately equal to the true ground truth probability distribution. Results by Woodworth et al. (2017), as discussed by Kilbertus et al. (2019), indicate that while fair prediction often leads to better performance than post-processing a (potentially) unfair predictor, the approximation $Q(y = 1 \mid \boldsymbol{x}, s) = P(y = 1 \mid \boldsymbol{x}, s) - \delta_s$ will "usually be suboptimal in terms of both utility and fairness".

3.2 Predicting vs. Deciding

While subdividing the problem of making decisions into predicting and deciding is a common approach in the field, recently Kilbertus et al. (2019) have argued, that using deterministic threshold policies is fundamentally flawed when applied to consequential decision making tasks using imperfect data. More specifically the approach presented in the previous section assumes that the data that is used to fit the parameters θ of the predictive model Q_θ, consists of independently and identically drawn samples from the true ground truth distribution. Kilbertus et al. (2019) propose that this assumption does not hold for many real-world applications, as in many applications the data has not been sampled from the true ground truth distribution $P(\boldsymbol{x}, s, y)$ but instead from a weighted distribution

$$P_{\pi_{Q_0}} \propto P(y \mid \boldsymbol{x}, s)\pi_0(d = 1 \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s) \tag{3.5}$$

where π_0 is some initial decision policy that is employed while collecting the (training) data. Going back to the example of a bank, giving out loans: it is unlikely that a bank would give out loans to everyone, while collecting the initial training data, that will then be used to train a decision making system. The potential economic damage of giving loan indiscriminately to everyone for the bank makes this approach infeasible. More likely the bank will base their decisions on historical data about the success of given out loans. But these loans were not given out to everyone that applied, but instead were given out to applicants that fulfilled some set of criteria which make up an initial decision policy π_0.

Kilbertus et al. (2019) call this kind of ground truth distribution $P_{\pi_{Q_0}}$ a distribution that is *induced* by π_0. In this scenario the decision whether or not $y \sim P(y \mid \boldsymbol{x}, s)$ comes into existence is based on the decision $d \sim \pi_0$. In terms of the loan scenario given before this means, that a bank that is training a machine learning model using historic data, can only use data, that they have actually collected. That means they are limited to the subset of applicants that they accepted according to their initial set of criteria π_0, as they have no data for the applicants, that were rejected. Kilbertus et al. (2019) argue that in such a scenario "for error based learning algorithms under no fairness constraints, learning within deterministic threshold policies is guaranteed to fail." They specifically show that given the scenario as defined above the following proposition holds:

Proposition 1. *If there exists a subset $\mathcal{V} \subset \mathcal{X} \times \mathcal{S}$ of positive measure under P such that $P(y = 1 \mid \mathcal{V}) \geq c$ and $P_{\pi_0}(y = 1 \mid \mathcal{V}) < c$, then there exists a maximum $Q_0^* \in \mathcal{Q}$ of $v_{P_{\pi_0}}$ such that $v_P(\pi_{Q_0^*}) < v_P(\pi_{Q^*})$.*

where $v_P(\pi)$ is a performance function defined as follows:

$$v_P(\pi) = u_{P_{\pi_0}}(\pi) - \frac{\lambda}{2}(\mathcal{F}_P(\pi))^2 \tag{3.6}$$

This function is a transformation of the constrained optimization problem 3.2 into a target function of an unconstrained optimization via the penalty method, which will be discussed in more detail in section 3.4.

This result implies that if there exists a sub-population for which the probability of being part of the desirable prediction class under the true distribution is greater than the confidence threshold c but is smaller than c under the probability distribution induced by π_0, then there exists an optimal model Q_0^* trained on the data drawn from P_{π_0} which is worse than the true optimal model Q^* w.r.t. the utility function. Kilbertus et al. (2019) go even further by showing that under certain assumptions even "a sequence of deterministic threshold rules, where each threshold rule is of the form of equation (3.4) and its associated predictive model is trained using the data gathered through the deployment of previous threshold rules, fails to recover the optimal policy despite it being in the hypothesis class."

However, the authors argue that an optimal decision policy π^* can be learned directly from data generated by a distribution P_{π_0} that is induced by π_0 if the initial data collecting policy π_0 is what they call an *exploring policy*:

Definition 3.2.1. *An **exploring policy** according to Kilbertus et al. (2019) is a policy that has a greater than zero probability of a positive decision, meaning "$\pi_0(d = 1 \mid \boldsymbol{x}, s) > 0$, for any measurable subset of $\mathcal{V} \subset \mathcal{X} \times \mathcal{S}$ with positive probability under P." This means for any sub-population $\mathcal{V}$ where the ground truth distribution puts probability mass, meaning $P(y = 1 \mid \mathcal{V}) > 0$, an exploring policy π_0 has to put probability mass as well.*

To learn such exploring policy, one still has to take into account, that the training data has been sampled from the induced distribution P_{π_0} instead of the ground truth distribution. Thankfully inverse probability weighting as explained in section 3.3 allows the recovery of the true utility $u_P(\pi)$ and $\mathcal{F}_P(\pi)$ even when given data only from the distribution P_{π_0}. Using this fact Kilbertus et al. (2019) propose the following:

Proposition 2. *Let* Π *be the set of exploring policies and* $\pi_0 \in \Pi \backslash \{\pi^*\}$. *Then the optimal objective value is*

$$v(\pi^*) = \sup_{\pi \in \Pi \backslash \{\pi^*\}} \left\{ u_{P_{\pi_0}}(\pi, \pi_0) - \frac{c}{2} (\mathcal{F}_{P_{\pi_0}}(\pi, \pi_0))^2 \right\}$$

That means the optimal decision policy $\pi*$ can be learned by maximizing the slightly altered version $v_{P_{\pi_0}}(\pi, \pi_0)$ of aforementioned the performance function where $u_{P_{\pi_0}}(\pi, \pi_0)$ and $\mathcal{F}_{P_{\pi_0}}(\pi, \pi_0)$ are the utility and fairness functions respectively, that have been recovered by using inverse probability weighting.

3.3 Inverse probability weighting

Inverse probability weighting is a technique which allows to calculate statistics over a pseudo-population that is different from the population from which the data has been sampled. This approach is commonly used in medical trials, to "simulate what would have been observed if the variable (or variables in the vector) L had not been used to decide the probability of treatment" (Hernán & Robins 2020), where L are confounding variables. One of the first weighted estimators was an estimator of the expected value introduced by Horvitz & Thompson (1952). The authors propose to weigh each sample i drawn from a distribution P inversely to how probable it was to be drawn in the first place, according to P. Applying this idea to the setting of fair decision making Kilbertus et al. (2019) propose to weight the utility $u(\pi)$ (and the constraint function $\mathcal{F}(\pi)$ respectively) by the inverse probability of the sample being part of the training data, according to π_0, leading to the utility function $u(\pi, \pi_0)$:

$$u(\pi, \pi_0) = \mathbb{E}_{\substack{\boldsymbol{x}, y, s \sim P_{\pi_0}(\boldsymbol{x}, s, y) \\ d \sim \pi(d \mid \boldsymbol{x}, s)}} \left[\frac{d(y - c)}{\pi_0(d = 1 \mid x, s)} \right] \tag{3.7}$$

This weighted utility function $u(\pi, \pi_0)$ can be shown to be equivalent to the original utility function $u(\pi)$, by applying importance sampling to the original utility. Importance sampling is a general technique to recover statistics of a particular distribution, given samples drawn from a different distribution, by reweighing them. To apply importance sampling to the utility function, lets first write the utility in its integral form

$$
\begin{aligned}
u_P(\pi) &= \mathbb{E}_{\substack{x,y,s\sim P(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}}[d(y-c)] \\
&= \int d(y-c)\pi(d \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s, y)\, dx\, dy\, ds\, dd
\end{aligned}
$$

Importance sampling can then be applied by multiplying the term $\frac{P_{\pi_0}(\boldsymbol{x},s,y)}{P_{\pi_0}(\boldsymbol{x},s,y)}$ to this integral, which is equivalent to multiplying by 1. After rearranging some of the terms the following new formulation of the utility function emerges

$$
\begin{aligned}
u_P(\pi) &= \int d(y-c)\pi(d \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s, y)\frac{P_{\pi_0}(\boldsymbol{x}, s, y)}{P_{\pi_0}(\boldsymbol{x}, s, y)}\, dx\, dy\, ds\, dd \\
&= \int d(y-c)\pi(d \mid \boldsymbol{x}, s)P_{\pi_0}(\boldsymbol{x}, s, y)\frac{P(\boldsymbol{x}, s, y)}{P_{\pi_0}(\boldsymbol{x}, s, y)}\, dx\, dy\, ds\, dd \\
&= \int \frac{P(\boldsymbol{x}, s, y)}{P_{\pi_0}(\boldsymbol{x}, s, y)} d(y-c)\pi(d \mid \boldsymbol{x}, s)P_{\pi_0}(\boldsymbol{x}, s, y)\, dx\, dy\, ds\, dd \\
&= \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[\frac{P(\boldsymbol{x}, s, y)}{P_{\pi_0}(\boldsymbol{x}, s, y)} d(y-c)\right] \\
&= \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[w(\boldsymbol{\omega})d(y-c)\right] = u_{P_{\pi_0}}(\pi, \pi_0)
\end{aligned}
$$

where the weights $w(\boldsymbol{\omega})$ can be interpreted as the following ratio, defined by Bottou et al. (2013):

$$
w(\boldsymbol{\omega}) = \frac{P(\boldsymbol{\omega})}{P_{\pi_0}(\boldsymbol{\omega})} = \frac{\text{factors appearing in } P(\boldsymbol{\omega}) \text{ but not in } P_{\pi_0}(\boldsymbol{\omega})}{\text{factors appearing in } P_{\pi_0}(\boldsymbol{\omega}) \text{ but not in } P(\boldsymbol{\omega})}
$$

As the true ground truth distribution P and the induced distribution P_{π_0} by definition share most of their terms, except for $\pi_0(d = 1 \mid \boldsymbol{x}, s)$, the weights $w(\boldsymbol{\omega})$ simplify to

$$
w(\boldsymbol{\omega}) = \frac{P(\boldsymbol{\omega})}{P_{\pi_0}(\boldsymbol{\omega})} = \frac{P(\boldsymbol{x}, s, y)}{P_{\pi_0}(\boldsymbol{x}, s, y)} = \frac{P(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)}{P(y \mid \boldsymbol{x}, s)\pi_0(d = 1 \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)} = \frac{1}{\pi_0(d = 1 \mid \boldsymbol{x}, s)}
$$

which are exactly the same, as the inverse probability weights, proposed by Kilbertus et al. (2019), therefore establishing the equivalence between $u_P(\pi)$ and $u_{P_{\pi_0}}(\pi, \pi_0)$. The same can under equivalent arguments be applied to fairness function $\mathcal{F}$ as well.

The problem with the weight choice of $\frac{1}{\pi_0(d=1|x,s)}$ is, that for samples that have been drawn from the distribution π_0 with a very small probability, the weights become very large, introducing numerical instability into the training process. For that reason Hernán & Robins (2020) introduce what they call stabilized IP weights. They propose to stabilize the weights, by choosing a different numerator for the fraction $\frac{1}{\pi_0(d=1|x,s)}$. The authors proof that the choice of weights $\frac{1}{\pi_0(d=1|x,s)}$ is a special case of

the weight choice $\frac{p}{\pi_0(d=1|x,s)}$ with $0 < p \leq 1$. More specifically they argue that "The stabilized weights $\frac{f[A]}{f[A|L]}$ are part of the larger class of stabilized weights $\frac{g[A]}{f[A|L]}$,where $g[A]$ is any function of A that is not a function of L." Hernán & Robins (2020) While the decision making policy is defined as $\pi(d = 1 \mid \boldsymbol{x}, s)$, this thesis refrains from using s as an input for π as to prevent disparate treatment (see Zafar et al. (2019)). For that reason the only confounding variable L as defined by Hernán & Robins (2020) is x. For that reason, in this thesis we chose $p = P(s \mid d = 1)$ as it exhibits the best performance empirically when applied to the algorithms discussed in chapter 4.

3.4 Penalty method

As talked about so far the goal of fair decision making is solving the constrained optimization problem 3.2 which means maximizing some utility function $u_{P_{\pi_0}}(\pi_\theta, \pi_0)$ under some fairness constraint $\mathcal{F}_{P_{\pi_0}}(\pi_\theta, \pi_0)$. The question then arises on how to solve this kind of constrained optimization problem efficiently. For a large number of problems efficient algorithms to solve constrained optimization problems are well known and explored, if certain assumptions about both the optimization target, as well as the constraints can be made. One such example would be linear programming, where efficient algorithms for solving the above stated optimization problem are known, under the assumption, that both the optimization target as well as the constraints are all linear. In the case of fair decision making this assumption does not hold, and indeed even the weaker notion of convexity cannot necessarily be assumed for the utility or the constraint function.

For that exact reason gradient optimization methods are often used in general machine learning applications to minimize a specified loss function, or equally maximize a specific utility. Given a optimization problem $\text{argmax}_\theta\, u_P(\pi_\theta)$ a gradient optimization methods such as gradient ascent will find a local maximum, given no further assumptions about $u_{P_{\pi_0}}(\pi_\theta)$ other than differentiability. But to apply such an algorithm, the optimization problem has to be given in an unconstrained form. A simple, intuitive approach to rewriting the constrained problem, as an unconstrained one, is the so called penalty method, which is the method chosen by Kilbertus et al. (2019). The penalty methods transforms any constrained problem of the form

$$\begin{aligned} \max_\theta \quad & f(x) \\ s.t. \quad & h_i(x) = 0, \forall i = 1, \ldots, p \end{aligned} \tag{3.8}$$

into an unconstrained problem, by adding a penalty term yielding the unconstrained optimization problem

$$\max_\theta f(x) - \rho \sum_{i=1}^{p} g(h_i(x)) \tag{3.9}$$

where $g(x)$ is the exterior penalty function. Kilbertus et al. (2019) chose the quadratic penalty function $g(x) = x^2$ and $\rho = \frac{\lambda}{2}$ for convenience when differentiating. The main disadvantage of this approach is described by Platt & Barr (1988):

> "Second, as more constraints are added, the constraint strengths get harder to set, especially when the size of the network (the dimensionality of x) gets large. In addition, there is a dilemma to the setting of the constraint strengths. If the strengths are small, then the system finds a deep local minimum, but does not fulfill all the constraints. If the strengths are large, then the system quickly fulfills the constraints, but gets stuck in a poor local minimum."

Kilbertus et al. (2019) skirt this issue by providing their experiment results across a logarithmic range of different choices for the cost constant λ.

3.5 Benefit function and benefit difference

So far this thesis has refrained from specifying a concrete fairness function $\mathcal{F}$ used to constrain the optimization problems formulated in equation (3.3) and equation (3.2). The reason for this is that fairness can be defined in multiple different ways, as seen in 2.1. Therefore enforcing different fairness criteria requires different fairness constraint functions. In this section the benefit difference, the fairness constraint chosen by Kilbertus et al. (2019), is going to be introduced. Firstly, lets recall the measures of group fairness, introduced in section 2.1: More specifically remember the definitions of demographic parity and equality of opportunity given as follows for the predictive setting:

- **Demographic parity:** $Q(\hat{y} = 1 \mid s = 0) = Q(\hat{y} = 1 \mid s = 1)$
- **Equality of opportunity:** $Q(\hat{y} = 1 \mid y = 1, s = 0) = Q(\hat{y} = 1 \mid y = 1, s = 1)$

where Q is some predictive model that makes some prediction $\hat{y}$. These formulations can be transferred to the decision making framework discussed in chapter 3 as follows:

- **Demographic parity:** $\int_x \pi(d = 1 \mid \boldsymbol{x}, s = 0) P(\boldsymbol{x} \mid s = 0)\, dx = \int_x \pi(d \mid \boldsymbol{x}, s = 1) P(\boldsymbol{x} \mid s = 0)\, dx$
- **Equality of opportunity:** $\int_x \pi(d = 1 \mid \boldsymbol{x}, s = 0) P(\boldsymbol{x} \mid y = 1, s = 0)\, dx = \int_x \pi(d = 1 \mid \boldsymbol{x}, s = 1) P(\boldsymbol{x} \mid y = 1, s = 1)\, dx$

To integrate these fairness constraints into their framework, Kilbertus et al. (2019) propose a formulation of these constraints as an expectation over the ground truth distribution and the decision policy, similar to the utility defined by (3.1). More specifically for the case of demographic parity they propose the following benefit function

$$b_P^s(\pi) = \mathbb{E}_{\substack{\boldsymbol{x} \sim P(\boldsymbol{x} \mid s) \\ d \sim \pi(d \mid \boldsymbol{x}, s)}} [d] \tag{3.10}$$

which, in the case of a binary decision variable, is equivalent to the definition of demographic parity given above, which can be shown as follows: Firstly the expectation over x and d can be written as two nested integrals:

$$\mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|s=0)\\ d\sim\pi(d|\boldsymbol{x},s=0)}}[d] = \mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|s=1)\\ d\sim\pi(d|\boldsymbol{x},s=1)}}[d]$$

$$\Leftrightarrow$$

$$\int_e \int_x d\pi(d \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid s=0)\, dd\, dx = \int_e \int_x d\pi(d \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid s=1)\, dd\, dx$$

Given the assumption that the decision variable is binary, meaning $e \in \{0, 1\}$, the above term simplifies to:

$$\int_x (0\pi(d=0 \mid \boldsymbol{x}, s=0) + 1\pi(d=1 \mid \boldsymbol{x}, s=0)) P(\boldsymbol{x} \mid s=0)\, dx\, dy = \int_x (0\pi(d=0 \mid \boldsymbol{x}, s=1) + 1\pi(d=1 \mid \boldsymbol{x}, s=1)) P(\boldsymbol{x} \mid s=1)\, dx\, dy$$

$$\Leftrightarrow$$

$$\int_x \pi(d=1 \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid s=0)\, dx\, dy = \int_x \pi(d=1 \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid s=1)\, dx\, dy$$

which is the exact definition of demographic parity. For the case of equality of opportunity Kilbertus et al. (2019) propose the benefit function $\mathbb{E}_{\substack{\boldsymbol{x},y\sim P(\boldsymbol{x},y|s=0)\\ d\sim\pi(d|\boldsymbol{x},s=0)}}[y \cdot d]$. This thesis proposes that this formulation actually does not represent equality of opportunity and that the following corrected version of the benefit function should instead be used:

$$b_P^s(\pi) = \mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|y=1,s)\\ d\sim\pi(d|\boldsymbol{x},s)}}[d] \tag{3.11}$$

With an equivalent argument as for demographic parity before, we can show that in the case of a binary decision variable this expectation is equivalent to the formulation of equality of opportunity:

$$\mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|y=1,s=0)\\ d\sim\pi(d|\boldsymbol{x},s=0)}}[d] = \mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|y=1,s=1)\\ d\sim\pi(d|\boldsymbol{x},s=1)}}[d]$$

$$\Leftrightarrow$$

$$\int_e \int_x d\pi(d \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid y=1, s=0)\, dd\, dx = \int_e \int_x d\pi(d \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid y=1, s=1)\, dd\, dx$$

As before the fact that $d \in \{0, 1\}$ can be used to simplify the formulation given above the same way as before:

$$\int_x 0\pi(d=0 \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid y=1, s=0) + 1\pi(d=1 \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid y=1, s=0)\, dx$$

$$=$$

$$\int_x 0\pi(d=0 \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid y=1, s=1) + 1\pi(d=1 \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid y=1, s=1)\, dx$$

$$\Leftrightarrow$$

$$\int_x \pi(d=1 \mid \boldsymbol{x}, s=0) P(\boldsymbol{x} \mid y=1, s=0)\, dx = \int_x \pi(d=1 \mid \boldsymbol{x}, s=1) P(\boldsymbol{x} \mid y=1, s=1)\, dx$$

which ends up to be the definition of equality of opportunity as defined before. But as mentioned already multiple times before in this chapter, the data is drawn from the distribution P_{π_0} induced by π_0 instead of the true ground truth distribution P. This leads to two issues: First of all, since the labels $yin\mathcal{Y}$ are only generated for samples which have been drawn from the distribution P_{π_0}, the distribution $P(\boldsymbol{x} \mid y=1, s=0)$ is unavailable. More specifically there exists no label information for data that has been rejected by the initial policy π_0. For that reason, this thesis will focus its exploration on demographic parity as the chosen measure of fairness.Secondly, even though the f-benefit $b^s_P(\pi)$ for demographic parity does not rely on the label information, it still has to be recovered via inverse propensity scoring equivalently to equation (3.7), such that $b^s_{P_{\pi_0}}(\pi, \pi_0) = \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}} \left[\frac{f(d,y)}{\pi_0(d=1|x,s)}\right]$.

Given the benefit-functions as defined above the a natural choice of the *benefit difference* as the choice of fairness function arises:

$$\mathcal{F}_{bf}(\pi, \pi_0) = b^0_P(\pi, \pi_0) - b^1_P(\pi, \pi_0) \tag{3.12}$$

.

Following equation (3.6), this choice of fairness function results in the overall optimization problem again using a penalty term to enforce the fairness constraint:

$$v_{P_{\pi_0}}(\pi, \pi_0) = u_{P_{\pi_0}}(\pi, \pi_0) - \frac{\lambda}{2}(b^0_{P_{\pi_0}}(\pi, \pi_0) - b^1_{P_{\pi_0}}(\pi, \pi_0))^2$$

3.6 Learning exploring policies

In Kilbertus et al. (2019) the authors demonstrate the practical application of proposition 2 in a practical setting. They restrict the parameterized class of exploring policies $\Pi(\Theta)$ to the family of logistic functions, meaning they choose, i.e.

$$\pi_\theta(d = 1 \mid \boldsymbol{x}, s) = \sigma(\phi(\boldsymbol{x}, s)^T\theta) = \frac{1}{1 + e^{-\phi(\boldsymbol{x},s)^T\theta}}$$

where $\theta \in \Theta \subset \mathbb{R}^m$ are the model parameters and $\phi : \mathbb{R}^d \times 0, 1 \to \mathbb{R}^m$ is a fixed feature map that maps the attributes of an individual into the parameter space. They solve the unconstrained optimization problem $v_P(\pi, \pi_0)$ by using stochastic gradient ascent on the model parameters θ. That means they iteratively update the model parameters according to the update rule $\theta_{t+1} = \theta_t + \alpha_t \nabla_{\theta_t} v_P(\pi_{\theta_t})$. The gradient of the unconstrained optimization problem with regards to θ_t can then be calculated as follows:

$$\begin{aligned}
\nabla_\theta v_P(\pi_\theta, \pi_0) &= \nabla_\theta \left(u(\pi_\theta, \pi_0) - \frac{\lambda}{2}(\mathcal{F}(\pi_\theta, \pi_0))^2 \right) \\
&= \nabla_\theta u(\pi_\theta, \pi_0) - \nabla_\theta \frac{\lambda}{2}(\mathcal{F}(\pi_\theta, \pi_0))^2 \\
&= \nabla_\theta u(\pi_\theta, \pi_0) - \lambda \mathcal{F}(\pi_\theta, \pi_0) \nabla_\theta \mathcal{F}(\pi_\theta, \pi_0)
\end{aligned}$$

Therefore to derive the overall gradient with regards to θ, the gradient of both the utility as well as the fairness function with regard to theta need to be derived first. To allow for the gradient of the utility to be expressed in terms of an expectation, Kilbertus et al. (2019) apply the log-derivative trick introduced in Williams (1992) which leads to the following derivation

$$
\begin{aligned}
\nabla_\theta u_{P_{\pi_0}}(\pi_\theta, \pi_0) &= \nabla_\theta \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y) \\ d\sim\pi_\theta(d|\boldsymbol{x},s)}} \left[\frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \right] \\
&= \nabla_\theta \int \frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \pi_\theta(d \mid \boldsymbol{x},s) P_{\pi_0}(\boldsymbol{x},s,y)\, dx\, dy\, ds\, dd \\
&= \int \frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \nabla_\theta \pi_\theta(d \mid \boldsymbol{x},s) \frac{\pi_\theta(d \mid \boldsymbol{x},s)}{\pi_\theta(d \mid \boldsymbol{x},s)} P_{\pi_0}(\boldsymbol{x},s,y)\, dx\, dy\, ds\, dd \\
&= \int \frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \pi_\theta(d \mid \boldsymbol{x},s) \frac{\nabla_\theta \pi_\theta(d \mid \boldsymbol{x},s)}{\pi_\theta(d \mid \boldsymbol{x},s)} P_{\pi_0}(\boldsymbol{x},s,y)\, dx\, dy\, ds\, dd \\
&= \int \frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \pi_\theta(d \mid \boldsymbol{x},s) \nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x},s) P_{\pi_0}(\boldsymbol{x},s,y)\, dx\, dy\, ds\, dd \\
&= \int \frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x},s) \pi_\theta(d \mid \boldsymbol{x},s) P_{\pi_0}(\boldsymbol{x},s,y)\, dx\, dy\, ds\, dd \\
&= \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y) \\ d\sim\pi_\theta(d|\boldsymbol{x},s)}} \left[\frac{d(y-c)}{\pi_0(d=1 \mid x,s)} \nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x},s) \right]
\end{aligned}
\tag{3.13}
$$

which can be applied equivalently for the fairness function $\mathcal{F}(\pi_\theta)$. Given the fact that $\pi_\theta \in \Pi(\Theta)$, the gradient of the logarithm of the policy with regards to the parameters $\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x}, s)$ can be derived analytically as well. The gradient $\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x}, s)$ given as follows:

$$
\begin{aligned}
\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x},s) &= \frac{\partial}{\partial \theta} \log\left(\sigma(\phi_i^T \theta)\right) \\
&= \frac{\partial}{\partial \theta} \log\left(\frac{1}{1+e^{-\phi_i^T \theta}}\right)
\end{aligned}
$$

can be rewritten using the chain rule as

$$
\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x},s) = \frac{\partial \log(\gamma)}{\partial \gamma} \frac{\partial \beta^{-1}}{\partial \beta} \frac{\partial e^{-\alpha}}{\partial \alpha} \frac{\partial \phi_i^T \theta_t}{\partial \theta_t}
$$

where α, β and γ are defined as

$$
\begin{aligned}
\alpha &:= -\phi_i^T \theta \\
\beta &:= 1 + e^{-\alpha} \\
\gamma &:= \frac{1}{\beta}
\end{aligned}
$$

Given this reformulation of the problem, $\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x}, s)$ is derived as follows:

$$\begin{aligned}
\nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x}, s) &= \frac{\partial \log(\gamma)}{\partial \gamma} \frac{\partial \beta^{-1}}{\partial \beta} \frac{\partial e^{-\alpha}}{\partial \alpha} \frac{\partial \phi_i^T \theta}{\partial \theta} \\
&= \frac{1}{\gamma} \left(-\frac{1}{\beta^2} \right) \left(-e^{-\alpha} \right) \phi_i \\
&= \frac{1}{\frac{1}{1+e^{-\phi_i^T \theta}}} \left(-\frac{1}{(1+e^{-\phi_i^T \theta})^2} \right) \left(-e^{-\phi_i^T \theta} \right) \phi_i \\
&= \left(-\frac{\left(1+e^{-\phi_i^T \theta}\right)}{(1+e^{-\phi_i^T \theta})^2} \right) \left(-e^{-\phi_i^T \theta} \right) \phi_i \\
&= \frac{e^{-\phi_i^T \theta}}{1+e^{-\phi_i^T \theta}} \phi_i = \frac{1}{1+e^{\phi_i^T \theta}} \phi_i = \frac{\phi_i}{1+e^{\phi_i^T \theta}}
\end{aligned}$$

where $\phi_i := \phi(\boldsymbol{x}_i, s_i)$ is the fixed feature map evaluated for a given sample i. The gradient of the utility function $\nabla_{\theta_t} u(\pi_\theta, \pi_0)$ for the parameter update of the policy at time step t can be estimated via Monte Carlo sampling from the policy at time step $t-1$ as:

$$\begin{aligned}
\nabla_{\theta_t} u_P(\pi_{\theta_t}, \pi_{\theta_{t-1}}) &\approx \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \frac{e_i(y_i - c)}{\pi_{\theta_{t-1}}(d=1 \mid \boldsymbol{x}_i, s_i)} \nabla_{\theta_t} \log \pi_{\theta_t}(d \mid \boldsymbol{x}_i, s_i) \\
&= \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \frac{e_i(y_i - c)}{\sigma(\phi_i^T \theta_{t-1})} \frac{\phi_i}{1+e^{\phi_i^T \theta_t}} \\
&= \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \frac{1}{\frac{1}{1+e^{-\phi_i^T \theta_{t-1}}}} e_i(y_i - c) \frac{\phi_i}{1+e^{\phi_i^T \theta_t}} \\
&= \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \frac{1+e^{-\phi_i^T \theta_{t-1}}}{1+e^{\phi_i^T \theta_t}} d_i(y_i - c) \phi_i
\end{aligned} \tag{3.14}$$

The gradient for the fairness function $\nabla \mathcal{F}$ can be derived using an equivalent argument. Kilbertus et al. (2019) propose algorithm 1 to train a decision making policy π_θ given training data that is drawn from a ground truth distribution P_{π_0} induced by an initial decision policy π_0, which they call *consequential learning*.

3.7 Empirical Results

3.7.1 Experimental Setup

In this section the performance of the `ConsequentialLearning` algorithm proposed by Kilbertus et al. (2019) will be evaluated on the three different datasets specified in section 2.2. This is partially a reproduction of the results by Kilbertus et al. (2019) as they released empirical results of their algorithm for the COMPAS dataset in their paper. The evaluation on the other two datasets represents new information and all of these results can be seen as a baseline of understanding for the extensions that will be introduced in chapter 4. To test the performance of the proposed algorithm, it will be tested similarly as proposed by Kilbertus et al. (2019), where they run training of the policy over a fixed number of time steps T where they collect data at each time step t. This simulates an online learning scenario, where a decision maker like a bank makes decisions in each time step t according to the policy that has been trained in the previous time step $t-1$ and then retrains the current policy based on the results of the accepted individuals. This setup give rise to two distinct strategies of using the data collected at each time step:

1. Train on only the data collected in the current time step t, which from now on will be denoted as D_t
2. Train on the entire history of data collected up to the current time step t, which from now on will be denoted as $D_{\leq t}$

All plots in this section depict the results of 30 independent training runs with $T = 50$ and the number of training points per time step as well as the size of the test set determined by the size of the overall dataset as described in section 2.2. The dark lines represent the median performance of the specified measure, while the shaded area around the median performance expresses the interquartile range, meaning the difference between the first and third quartile. Across all experiments the same set of seeds have been used to ensure comparability of the results. As discussed in section 3.6, the algorithm used to train the policies in these experiments will be stochastic gradient descent. The parameter settings that have been used to achieve the results presented in this section can be found in table 6.1.

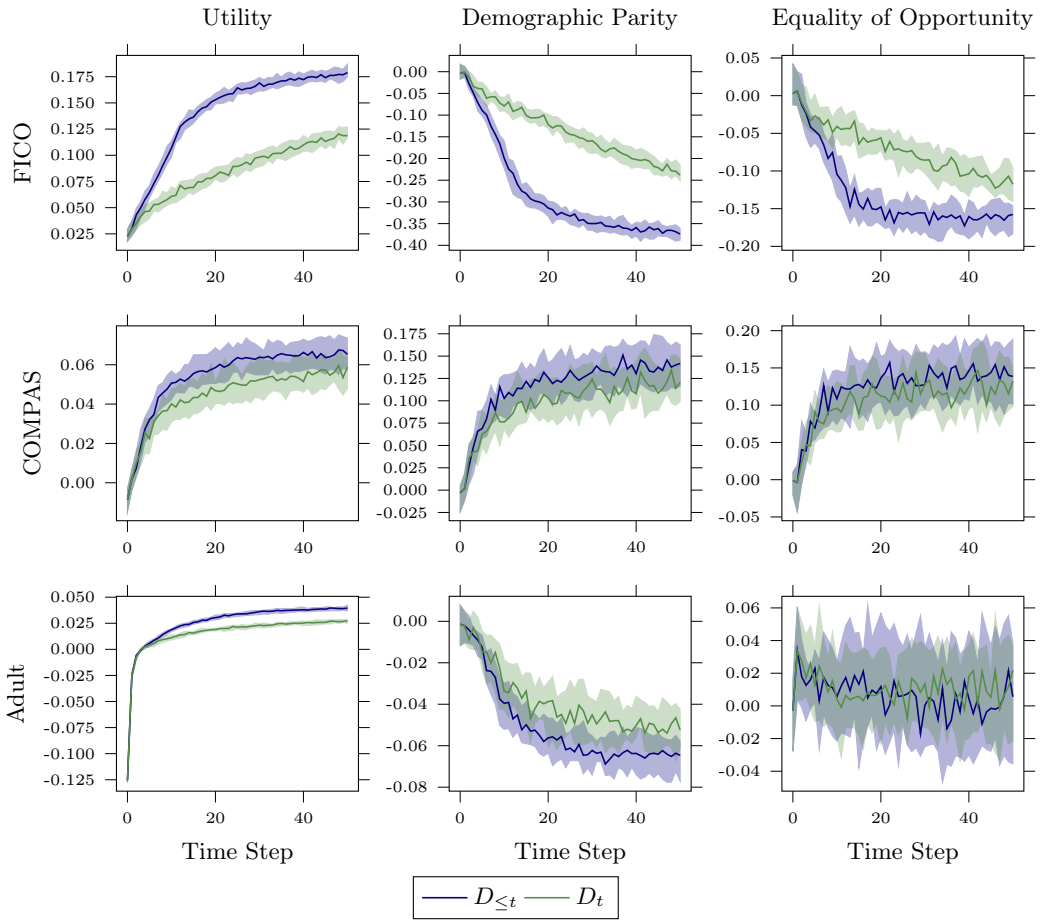

Figure 3.1: The performance of the algorithm across all datasets over $t \in [0, 50]$ time steps, while training unconstrained, meaning $\lambda = 0$. The first row represents the performance on the FICO dataset, the second row on the COMPAS dataset and the last row on the adult dataset.

3.7.2 Results

First off figure 3.1 shows the performance of the fair decision making framework without any fairness constraint, on all three datasets, meaning the fairness constant λ was set to 0. As one can see, the utility increases steadily over time, indicating that given an exploring policy π_0 some (locally) optimal policy can potentially be found. As might be expected the setting where the policy is trained on the entire history $D_{\leq t}$ converges faster than the strategy where only the data of the current time step is used. But when monitoring the fairness measures at the same time it becomes obvious that this solution is unfair in the sense that both in terms of demographic parity and equality of opportunity one group is favoured significantly.

For that reason some fairness constraint $\mathcal{F}$ has to be enforced, by choosing a fitting fairness constant λ. The penalty method used by Kilbertus et al. (2019) has no systematic way of choosing λ, as described in section 3.4. The authors therefore decided to evaluate the performance of their algorithm over a range of different values of λ. The following plots show the performance of the trained policy π at the final time step T for all values of the fairness constant.

A common trend that can be observed across all the evaluated datasets is that at some point across the range of λ there is a drop in both utility and unfairness. This drop signifies the point where the policy π has degenerated to a point, where it rejects all or at least most applicants. The question, that will be answered in the following chapters is, whether a fairness constant λ can be found, that both maintains a degree of utility while at the same time achieving fairness. Additionally it can be observed, that for both COMPAS and the adult datasets the results are quite noisy. And as with the unconstrained case the training on the entire history of data, instead of just the data received in the current time step t is slightly better behaved in terms of maximum utility as well as general noisyness of the results.

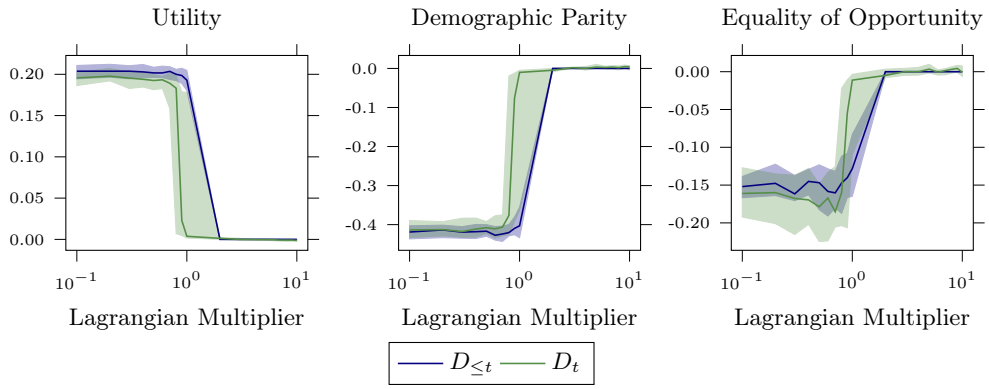

Figure 3.2: The performance evaluated on the FICO dataset when enforcing demographic parity via the benefit difference as the fairness function $\mathcal{F}$. The range of the fairness constant λ was chosen to be $(10^{-1}, 10^{1})$. As with all the experiments, a steep drop in both utility and fairness can be observed, in this case around $\lambda \approx 1$. The setting where the policy is trained on the entire data collected in previous time steps $D_{\leq t}$ instead of just the data collected in the current time step D_t, is slightly less noisy, especially for larger values of λ.

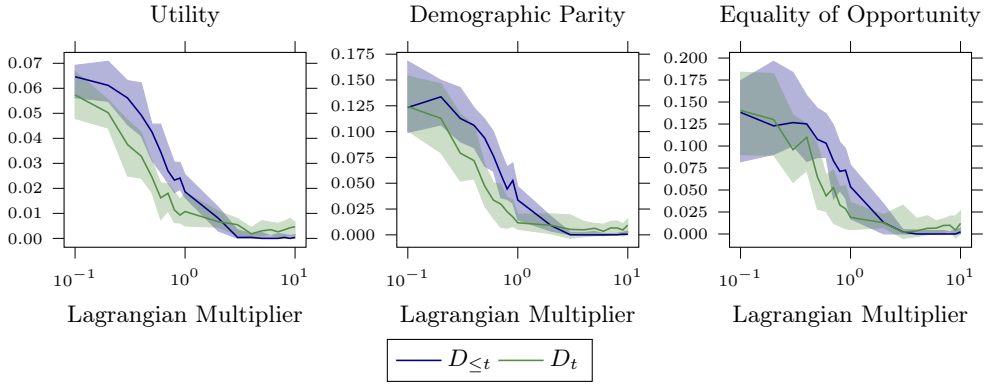

Figure 3.3: The performance evaluated on the COMPAS dataset when enforcing demographic parity via the benefit difference as the fairness function $\mathcal{F}$. The range of the fairness constant λ was chosen to be $(10^{-1}, 10^{1})$. As with all the experiments, a steep drop in both utility and fairness can be observed, but the drop is more gradual than in the case of the FICO data set. The setting where the policy is trained on the entire data collected in previous time steps $D_{\leq t}$ instead of just the data collected in the current time step D_t, is slightly less noisy, especially for larger values of λ. Overall enforcing demographic parity seems to lead to significantly less noisy results.

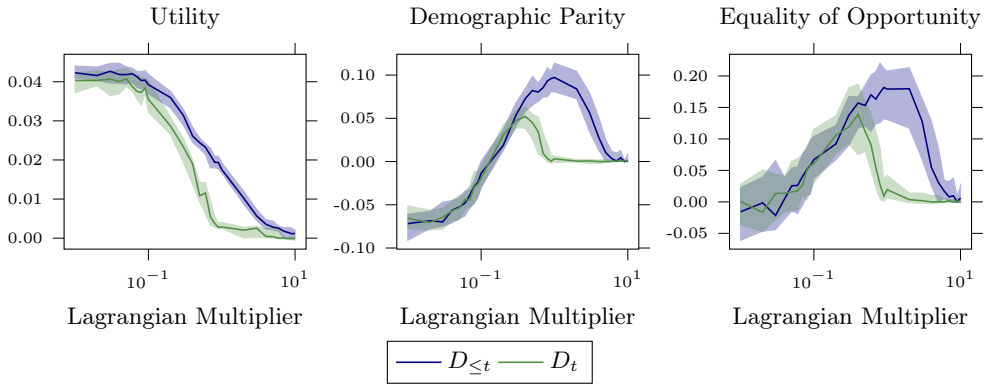

Figure 3.4: The performance evaluated on the adult dataset when enforcing demographic parity via the benefit difference as the fairness function $\mathcal{F}$. The range of the fairness constant λ was chosen to be $(10^{-2}, 10^{1})$. On the adult dataset the algorithm displays significantly different behaviour than on the other two: While enforcing equality of opportunity yields similar results as on the other dataset. Enforcing demographic parity on the other hand leads to increasing inequity when increasing λ, before finally dropping off for a large enough lambda, all the while decreasing utility in a more linear fashion.

4 Optimizing the utility-fairness trade-off

So far this thesis has focused on an introduction to fair decision making based on the results of Kilbertus et al. (2019). In this chapter the trade-off between utility and fairness in fair decision making will be explored further. Multiple natural extension to the approaches implemented by Kilbertus et al. (2019) will be presented as the main contributions made by this thesis.

As seen in section 3.7, both the utility as well as the fairness functions exhibit a relatively large amount of noise for a chosen range of the fairness parameter λ, especially in the more complex, higher-dimensional problem settings. Additionally the fairness constraint function displays non-smooth behaviour, especially for small values of λ. As one of the goals of this thesis, is to present a gradient based algorithmic solution for training λ, smoother and less noisy behaviour of both utility and fairness function is desirable. For this reason section 4.1 will introduce multiple different measures taken to improve the overall behaviour of the optimization, such as using ADAM instead of SGD as well as a relaxation of the fairness constraint inspired by Zafar et al. (2019). Following that, the sections 4.2 and 4.3 will discuss algorithmic approaches to chose the trade-off parameter λ. In Section 3.4 the quadratic penalty method used by Kilbertus et al. (2019) to solve the constrained optimization problem 3.2 has been discussed. As mentioned in this chapter, one of the main challenges of this approach is choosing a fitting value for the fairness parameter λ. For that reason this thesis is going to pivot away from the penalty method and will instead explore approaches of transforming problem 3.2 into an unconstrained optimization problem using the theory of Lagrangian multipliers and the duality property of optimization problems. These considerations will lead to two concrete algorithmic solutions called `DualConsequentialLearning` and `AugmentedDualConsequentialLearning`, that allow for the training of both the model parameters θ as well as the Lagrangian multiplier λ at the same time.

4.1 Improving optimization procedure

4.1.1 Optimization algorithm: SGD vs. ADAM

As discussed in the introduction to this chapter, the first goal of this section is to improve the optimization procedure used to train the parameters. Kilbertus et al. (2019) propose to train the policy parameters by using the default stochastic gradient

method. Prior works such as Dogo et al. (2018) have shown empirically that for many use cases vanilla SGD gets outperformed by newer algorithms proposed in recent years. One such algorithm is called adaptive moment estimation (ADAM) which was first proposed by Kingma & Ba (2014). The main goal of ADAM is to dynamically adapt the learning rate, based on estimates of the first and second order moments of the gradient. The authors estimate the first and second moment of the gradient with the following moving averages

$$\begin{aligned} g_t &= \nabla_{\theta_t} v_P(\pi_{\theta_t}) \\ m_t &= \beta_1 \cdot m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 \cdot v_{t-1} + (1-\beta_2) g_t^2 \end{aligned}$$

where g_t denotes the gradient, and β_1, β_2 are the parameters that "control the decay rates of these moving averages". The authors argue that, given an initialization of m and v to (vectors of) zero, biases the moving averages towards zero. To counteract this effect, they bias-correct the moment estimates as follows:

$$\begin{aligned} \hat{m}_t &= m_t / (1-\beta_1^t) \\ \hat{v}_t &= v_t / (1-\beta_2^t) \end{aligned}$$

Instead of updating the model parameters θ with the update rule of the vanilla stochastic gradient method

$$\theta_{t+1} = \theta_t + \alpha_t \nabla_{\theta_t} v_P(\pi_{\theta_t})$$

the update rule of the model parameters changes to

$$\theta_{t+1} = \theta_t + \alpha_t \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where ϵ is a very small value that ensures that divisions by zero don't occur. Kingma & Ba (2014) suggest $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ as default values for the newly introduced hyper-parameters. Figure 4.1 shows the improvement of ADAM compared to vanilla stochastic gradient methods on the FICO dataset. This example shows, that especially given the training scenario D_t, which is the more computationally efficient one, where only the data collected in the current time step is used for training, profits from using ADAM in terms of its convergence rate.

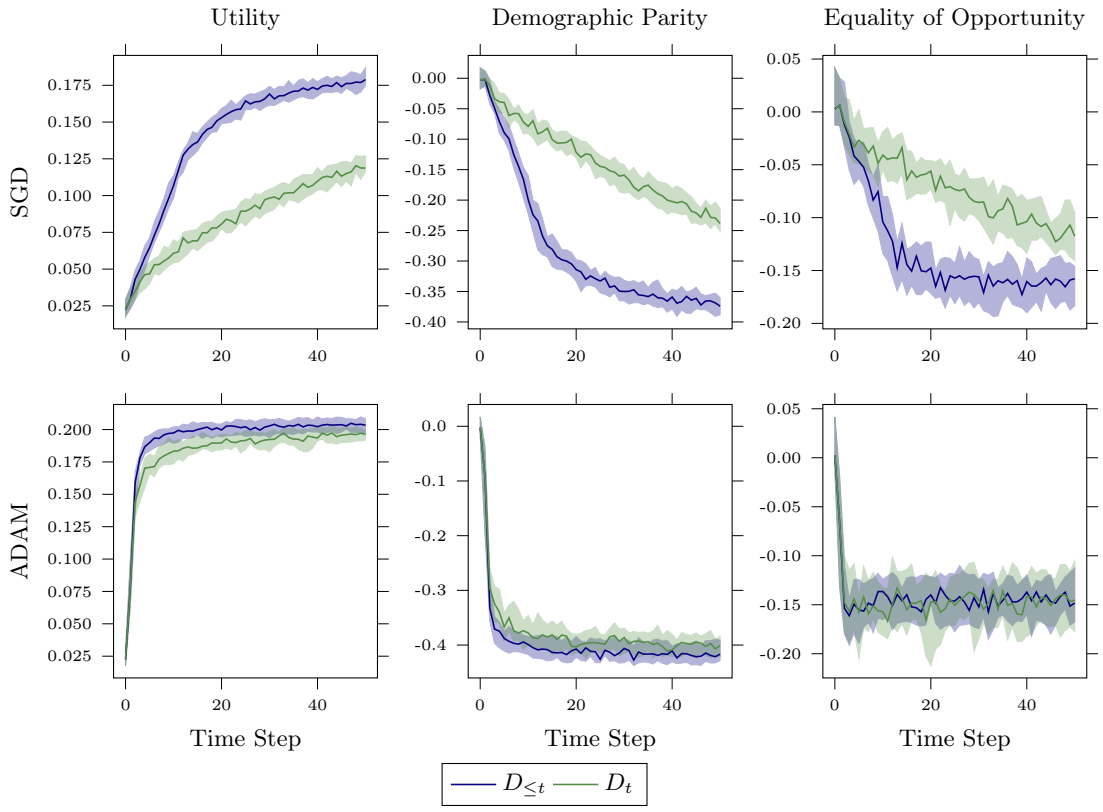

Figure 4.1: Comparison of stochastic gradient descent with ADAM on the FICO dataset for $\lambda = 0$: For both the scenario where the policy is trained on the entire history $D_{\leq t}$ as well as the where training is only executed on the data collected in the current time step D_t ADAM converges much faster than SGD. But especially for D_t covergence improves drastically. This is expected as the momentum terms m_t and v_t that are being kept by ADAM preserve gradient information across time steps, and therefore store information about the entire dataset.

4.1.2 Relaxed Fairness constraint: Covariance of decision

To tackle the non-smooth behaviour of the benefit difference function, especially for smaller choices of the fairness parameter λ as discussed in the intro to this chapter, which can also be seen in figure 4.2, this chapter will now introduce a different, relaxed notion of fairness inspired by Zafar et al. (2019). They, in the context of fair classification, propose to formulate the fairness constraint in terms of the covariance of the sensitive attribute s of an individual and the signed distance of that individuals non-sensitive feature vector $\boldsymbol{x}$ from the decision boundary. They define that distance in terms of a distance function which from now on will be referenced as $D_\theta(\boldsymbol{x}, y)$. They measure the covariance between $D_\theta(\boldsymbol{x}, y)$ and the sensitive attribute s as:

$$Cov_{DI}(s, d_\theta(\boldsymbol{x})) = \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mathbb{E}_{s\sim P(s)}[s])(D_\theta(\boldsymbol{x}, y) - \mathbb{E}_{\boldsymbol{x},y\sim P(\boldsymbol{x},y)}[D_\theta(\boldsymbol{x}, y)])\right]$$

Which can be simplified in the following way

$$\begin{aligned}
Cov_{DI}(s, D_\theta(\boldsymbol{x}, y)) =& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mathbb{E}_{s\sim P(s)}[s])(D_\theta(\boldsymbol{x}, y) - \mathbb{E}_{\boldsymbol{x},y\sim P(\boldsymbol{x},y)}[D_\theta(\boldsymbol{x}, y)])\right] \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)(D_\theta(\boldsymbol{x}, y) - \mu_{D_\theta})\right] \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y) - (s - \mu_s)\mu_{D_\theta}\right] \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y)\right] - \mathbb{E}_{s\sim P(s)}\left[(s - \mu_s)\mu_{D_\theta}\right] \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y)\right] - \mu_{D_\theta}\mathbb{E}_{s\sim P(s)}\left[(s - \mu_s)\right] \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y)\right] - \mu_{D_\theta}(\mathbb{E}_{s\sim P(s)}\left[s\right] - \mathbb{E}_{s\sim P(s)}\left[\mu_s\right]) \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y)\right] - \mu_{D_\theta}\underbrace{(\mu_s - \mu_s)}_{=0} \\
=& \mathbb{E}_{\boldsymbol{x},y,s\sim P(\boldsymbol{x},y,s)}\left[(s - \mu_s)D_\theta(\boldsymbol{x}, y)\right] \\
\approx& \sum_{\boldsymbol{x},s,y\sim\mathcal{D}} (s - \mu_s)D_\theta(\boldsymbol{x}, y)
\end{aligned} \tag{4.1}$$

where $\mathcal{D}$ is the data-set sampled from the ground truth distribution. The authors state that the overall fairness constraint function $\mathcal{F}_{cov}$ is convex in θ given the distance function $D_\theta(\boldsymbol{x}, y)$ is linear in θ. If $D_\theta(\boldsymbol{x}, y)$ is convex in θ then the authors show, that their constraint function $\mathcal{F}_{cov}$ can be written in such a way, that it is convex concave in θ.

In the decision making setting described in sections 3.2 and beyond there is no decision boundary as described by Zafar et al. (2019), but instead the idea of using the covariance as a fairness measure is extrapolated to the decisions d sampled from the decision policy π as follows:

$$Cov_{DI}(s, e) = \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s) \\ d\sim\pi(d|\boldsymbol{x},s)}}\left[(s - \mu_s)(d - \mu_d)\right] \tag{4.2}$$

This is expected to improve the smoothness of the optimization process. This expectation is justified intuitively by the comparison of the behaviour of this covariance based fairness function compared to the benefit difference, as seen in figure 4.2. It shows that the covariance of decision not only exhibits as lower variance, but it also behaves much more similarly to the utility both in shape and magnitude.

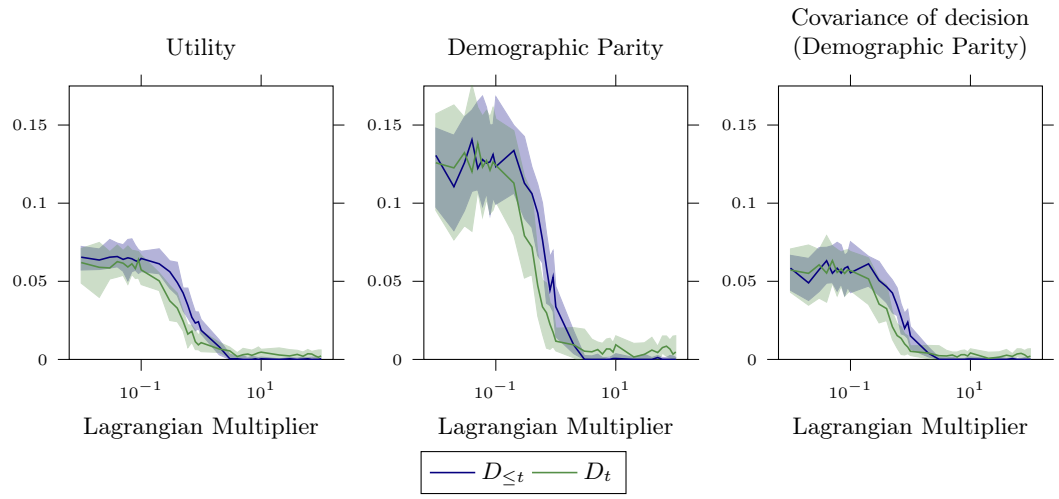

Figure 4.2: Comparison of the benefit difference and covariance of decision on the COMPAS dataset: In this scenario the fairness function used to constrain the optimization problem was demographic parity, but the results for the covariance of decisions was also tracked for every value of λ. Especially for smaller values of lambda, the covariance function shows less noisy behaviour. The average interquartile range across all time steps for the benefit difference was 0.027 for $D_{\leq t}$ and 0.035 for D_t while for the covariance of decision the respective IQRs were 0.012 and 0.015. This behaviour might prove useful when learning λ with a gradient based approach.

Additionally to this empirical justification, this thesis will now establish a theoretical connection between 4.2 and the difference of benefits discussed in section 3.5 to justify this choice of fairness constraint function. Equation 4.2 simplifies to

$$\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[(s-\mu_s)d\right]$$

under the equivalent argument as used in 4.1. This form can be further be rewritten using the rules governing expectations, as follows:

$$\begin{aligned}
\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[(s-\mu_s)d\right] &= \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[sd-\mu_s d\right] \\
&= \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[sd\right] - \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[\mu_s d\right] \\
&= \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[sd\right] - \mu_s\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[d\right] \\
&= \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[sd\right] - \mu_s\mu_d
\end{aligned}$$

As per the definition of an expectation we can rewrite the left hand side of the above given difference as the following integrals:

$$\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}[sd] = \int_x \int_e \int_s (s\cdot d)\pi(d \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\, dx\, dd\, ds$$

which in the case of a binary $s \in \{0, 1\}$ and $d \in \{0, 1\}$ further simplifies

$$\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s)\\ d\sim\pi(d|\boldsymbol{x},s)}}[sd] = \int_x \sum_{d\in\{0,1\}} \sum_{s\in\{0,1\}} (s\cdot d)\pi(d \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\, dx$$

As s is a label, the choice of $s \in 0, 1$ is an arbitrary one and can therefore be changed to $s \in -1, 1$. Using this fact the sum over all s can be written as

$$\begin{aligned}
&\int_x \sum_{d\in\{0,1\}} \sum_{s\in\{1,-1\}} (s\cdot d)\pi(d \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\, dx \\
=&\int_x \sum_{d\in\{0,1\}} (1\cdot d)\pi(d \mid \boldsymbol{x}, s=1)P(\boldsymbol{x}, s=1) \\
&+ (-1\cdot d)\pi(d \mid \boldsymbol{x}, s=-1)P(\boldsymbol{x}, s=-1)\, dx \\
=&\int_x \sum_{d\in\{0,1\}} d\pi(d \mid \boldsymbol{x}, s=1)P(\boldsymbol{x}, s=1) \\
&- d\pi(d \mid \boldsymbol{x}, s=-1)P(\boldsymbol{x}, s=-1)\, dx
\end{aligned}$$

The sums and integrals can then be split along the lines of group membership according to the sensitive attribute s

$$\begin{aligned}
=&\int_x \sum_{d\in\{0,1\}} d\pi(d \mid \boldsymbol{x}, s=1)P(\boldsymbol{x}, s=1)\, dx \\
&-\int_x \sum_{d\in\{0,1\}} d\pi(d \mid \boldsymbol{x}, s=-1)P(\boldsymbol{x}, s=-1)\, dx
\end{aligned}$$

and the joint probabilities $P(x, s = 1)$ and $P(x, s = -1)$ can be decomposed using the chain rule of probability:

$$\begin{aligned}
=&\int_x \sum_{d\in\{0,1\}} d\pi(d \mid \boldsymbol{x}, s=1)P(\boldsymbol{x} \mid s=1)P(s=1)\, dx \\
&-\int_x \sum_{d\in\{0,1\}} d\pi(d \mid \boldsymbol{x}, s=-1)P(\boldsymbol{x} \mid s=-1)P(s=-1)\, dx
\end{aligned}$$

This term can then be written in terms of the definition of the benefit for demographic parity as defined by (3.10)

$$\begin{aligned}
&= \mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|s=1) \\ d\sim\pi(d|\boldsymbol{x},s=1)}}[d] \cdot P(s=1) - \mathbb{E}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}|s=-1) \\ d\sim\pi(d|\boldsymbol{x},s=-1)}}[d] \cdot P(s=-1) \\
&= b_P^1 \cdot P(s=1) - b_P^{-1} \cdot P(s=-1)
\end{aligned}$$

which bears some resemblance to the benefit difference for demographic parity as defined in (3.12). Putting it all together, one ends up with an alternative expression for the covariance

$$\begin{aligned}
\mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s) \\ d\sim\pi(d|\boldsymbol{x},s)}}[(s-\mu_s)d] &= \mathbb{E}_{\substack{\boldsymbol{x},s\sim P(\boldsymbol{x},s) \\ d\sim\pi(d|\boldsymbol{x},s)}}[sd] - \mu_s\mu_d \\
&= b_P^1 \cdot P(s=1) - b_P^{-1} \cdot P(s=-1) - \mu_s\mu_d
\end{aligned}$$

which even further simplifies under more strict assumptions about the distribution $P(s)$. More specifically, if it is assumed that $P(s=1)$ and $P(s=-1)$ are the same, meaning equal proportions of the population have $s=1$ and $s=-1$.[1] Under this additional assumption it is obvious that $P(s=1) = P(s=-1) = \frac{1}{2}$ and $\mu_s = 0$, leading to the following simplification of the above statement:

$$b_P^1 \cdot \frac{1}{2} - b_P^{-1} \cdot \frac{1}{2} - 0\mu_d = \frac{1}{2}(b_P^1 - b_P^{-1})$$

This means that the covariance of the sensitive attribute s and the decision d under the assumptions that $s \in \{0,1\}$ and the population is balanced with regards to s can be written as the benefit difference $b_P^1 - b_P^{-1}$ scaled by a constant $c = \frac{1}{2}$. Therefore establishing an intuitive connection between the benefit difference and the relaxed covariance formulation of fairness. As before, the covariance formulation used above assumes data to be given by the true ground truth distribution P. Instead the data is generated by the induced distribution P_{π_0} as discussed in chapter 3. Therefore inverse propensity scoring is again, as with the benefit difference constraint before, used to recover the fairness constraint function $\mathcal{F}(\pi_\theta, \pi_0)$ from the data given by the induced ground truth distribution P_{π_0} instead of the true ground truth distribution P. In the case of the covariance formulation of $\mathcal{F}$ as given by 4.1 there are two separate expectations that have to be taken into account, when inverse propensity scoring. Specifically note that the mean of the sensitive attribute μ_s can be written as

$$\mu_s = \mathbb{E}_{s\sim P(s)}[s] = \int_s sP(s)\,ds$$

[1] This might seem as a strong assumption on first sight, but as the fairness constraint is only used during training time, this means only the training data needs to be balanced with regards to s. This is easily achieved by adjusting the Monte Carlo sampling process in such a way, that a sampled batch of data contains an equal amount of samples from each group.

which can be rewritten with regards to the overall joint distribution $P(\boldsymbol{x}, s, y)$ instead of the marginal distribution $P(s)$, by only using the sum and product rules of probability:

$$
\begin{aligned}
\mathbb{E}_{s\sim P(s)}[s] &= \int_s sP(s)\,ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y P(s \mid \boldsymbol{x}, y)P(y \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y \frac{P(s, \boldsymbol{x}, y)}{P(\boldsymbol{x}, y)}P(y \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y \frac{P(y \mid s, \boldsymbol{x})P(s \mid \boldsymbol{x})P(\boldsymbol{x})}{P(\boldsymbol{x}, y)}P(y \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y \frac{P(y \mid s, \boldsymbol{x})P(s \mid \boldsymbol{x})P(\boldsymbol{x})}{P(y \mid \boldsymbol{x})P(\boldsymbol{x})}P(y \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y \frac{P(y \mid s, \boldsymbol{x})P(s \mid \boldsymbol{x})}{P(y \mid \boldsymbol{x})}P(y \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y P(y \mid \boldsymbol{x}, s)P(s \mid \boldsymbol{x})P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y P(y \mid \boldsymbol{x}, s)\frac{P(\boldsymbol{x}, s)}{P(\boldsymbol{x})}P(\boldsymbol{x})\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s s\left(\int_{\boldsymbol{x}}\int_y P(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\,d\boldsymbol{x}\,dy\right)ds \\
&= \int_s\int_{\boldsymbol{x}}\int_y sP(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\,d\boldsymbol{x}\,dy\,ds = \mathbb{E}_{\boldsymbol{x},s,y\sim P(\boldsymbol{x},s,y)}[s]
\end{aligned}
$$

Given this formulation of the expected value of s under the true joint ground truth distribution $P(\boldsymbol{x}, s, y)$ we can further see that μ_s under P can be recovered from data given by the induced distribution P_{π_0} by applying inverse probability weighting, as discussed previously in section 3.3:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{x},s,y\sim P(\boldsymbol{x},s,y)}[s] &= \int_s\int_{\boldsymbol{x}}\int_y sP(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\,d\boldsymbol{x}\,dy\,ds \\
&= \int_s\int_{\boldsymbol{x}}\int_y sP(y \mid \boldsymbol{x}, s)P(\boldsymbol{x}, s)\frac{\pi_0(d=1 \mid x, s)}{\pi_0(d=1 \mid x, s)}\,d\boldsymbol{x}\,dy\,ds \\
&= \int_s\int_{\boldsymbol{x}}\int_y \frac{s}{\pi_0(d=1 \mid x, s)}P(y \mid \boldsymbol{x}, s)\pi_0(d=1 \mid x, s)P(\boldsymbol{x}, s)\,d\boldsymbol{x}\,dy\,ds \\
&= \int_s\int_{\boldsymbol{x}}\int_y \frac{s}{\pi_0(d=1 \mid x, s)}P_{\pi_0}(\boldsymbol{x}, s, y)\,d\boldsymbol{x}\,dy\,ds \\
&= \mathbb{E}_{s\sim P_{\pi_0}(\boldsymbol{x},s,y)}\left[\frac{s}{\pi_0(d=1 \mid x, s)}\right] = \mu_{s_{\pi_0}}
\end{aligned}
$$

Using the inverse propensity scored expectation of s the covariance of the sensitive attribute and the decision d can be recovered from data given by the induced ground truth distribution P_θ in a similar fashion as seen for both the utility and the benefit difference:

$$\begin{aligned}\mathcal{F}_{Cov_P(\pi)} &= \mathbb{E}_{\substack{x,y,s\sim P(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[(s-\mu_{s_{\pi_0}})d\right] \\ &= \mathbb{E}_{\substack{\boldsymbol{x},y,s\sim P_{\pi_0}(\boldsymbol{x},s,y)\\ d\sim\pi(d|\boldsymbol{x},s)}}\left[\frac{(s-\mu_{s_{\pi_0}})d}{\pi_0(d=1\mid x,s)}\right] = \mathcal{F}_{Cov_{P_{\pi_0}}}(\pi,\pi_0)\end{aligned}$$

4.2 The Lagrangian and Duality

To find a more principled approach to solving the constrained optimization problem 3.2 we can consult the theory of constrained optimization, and more specifically the Lagrange function. Lets first note that any maximization of some function f has an equivalently formulation as an minimization problem of the negative of this function. We can therefore write 3.2 as the following optimization problem instead

$$\begin{aligned}\underset{\theta}{\text{argmin}} \quad & -u_P(\pi_\theta) \\ s.t. \quad & \mathcal{F}_P(\pi_\theta) = 0\end{aligned} \tag{4.3}$$

which is called the standard form of a continuous optimization problem. The most general form of a continuous optimization problem in standard form, as specified in Boyd & Vandenberghe (2004), is

$$\begin{aligned}\min_x \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0, \forall i = 1,\dots,m \\ & h_i(x) = 0, \forall i = 1,\dots,p\end{aligned} \tag{4.4}$$

with $x \in \mathbb{R}^n$, which is called the primal problem P with an the joint, non-empty domain $\mathcal{D} = (\bigcap_{i=0}^m \mathbf{dom} f_i) \cap (\bigcap_{i=0}^p \mathbf{dom} h_i)$ containing the optimal solution $p*$. For such a problem P we can formulate the so called Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ with domain $L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$.

$$\mathcal{L}(x,\nu,\lambda) = f_0(x) + \sum_{i=1}^m \nu_i f_i(x) + \sum_{i=1}^p \lambda_i h_i(x) \tag{4.5}$$

The Lagrangian is the objective function $f_0(x)$ augmented with the sum of the constraint functions weighted by the so called Lagrangian multipliers ν and λ. Given the

Lagrangian $\mathcal{L}(x, \nu, \lambda)$ of the primal optimization problem P one can now define the Lagrangian dual function $g(\nu, \lambda)$:

$$g(\nu, \lambda) = \inf_{x \in \mathcal{D}} \mathcal{L}(x, \nu, \lambda) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \nu_i f_i(x) + \sum_{i=1}^{p} \lambda_i h_i(x) \right)$$

Given the dual function $g(\nu, \lambda)$ weak duality holds for all $f_0(x)$ under the assumption that $\nu \geq 0$, meaning for all feasible solutions $\tilde{x}$ of the original optimization problem P the following inequality holds

$$g(\nu, \lambda) \leq p*$$

This property is proven by Boyd & Vandenberghe (2004) as follows: Given any feasible point $\tilde{x}$ of the original optimization problem P with $\nu > 0$, the inequality

$$\sum_{i=1}^{m} \nu_i \underbrace{f_i(x)}_{\leq 0} + \sum_{i=1}^{p} \lambda_i \underbrace{h_i(x)}_{=0} \leq 0$$

holds and therefore

$$\mathcal{L}(\tilde{x}, \nu, \lambda) = f_0(\tilde{x}) + \sum_{i=1}^{m} \nu_i f_i(\tilde{x}) + \sum_{i=1}^{p} \lambda_i h_i(\tilde{x}) \leq f_0(\tilde{x})$$

Hence, the solution of $g(\nu, \lambda)$ is a lower bound for $p*$ as

$$g(\nu, \lambda) = \inf_{x \in \mathcal{D}} \mathcal{L}(x, \nu, \lambda) \leq \mathcal{L}(\tilde{x}, \nu, \lambda) \leq f_0(\tilde{x})$$

must hold for all feasible $\tilde{x}$, meaning also for $p*$ specifically. Furthermore it can be shown, as done in Sagnol (2017), that the following min-max inequality is a valid alternative formulation of the weak duality property:

$$\begin{aligned} \sup_{\substack{\nu \in \mathbb{R}_+^m \\ \lambda \in \mathbb{R}^p}} \inf_{x \in \mathcal{D}} \mathcal{L}(x, \nu, \lambda) &\leq \inf_{x \in \mathcal{D}} \sup_{\substack{\nu \in \mathbb{R}_+^m \\ \lambda \in \mathbb{R}^p}} \mathcal{L}(x, \nu, \lambda) \\ &\Leftrightarrow \\ \sup_{\substack{\nu \in \mathbb{R}_+^m \\ \lambda \in \mathbb{R}^p}} g(\nu, \lambda) &\leq p* \end{aligned} \tag{4.6}$$

From this formulation of weak duality, one can derive the Lagrangian dual D, which offers a simple way to calculate the lower bound of the original optimization problem P, as

$$\begin{aligned} \max_{\substack{\nu \in \mathbb{R}^m_+ \\ \lambda \in \mathbb{R}^p}} \quad & g(\nu, \lambda) \\ s.t. \quad & \nu_i \geq 0, \forall i = 1, \ldots, m \end{aligned} \tag{4.7}$$

A useful fact of D, is that it is always concave, even if f and $\mathcal{D}$ are non-convex. This is the case, as for any $x \in \mathcal{D}$ the Lagrangian $\mathcal{L}(x, \nu, \lambda)$ is a linear function of ν and λ. Therefore "$g(\nu, \lambda)$ is an infimum over linear functions hence concave."Hardt & Simchowitz (2018)

Given the fact that the constrained optimization problem in equation 4.3, that constitutes fair decision making, does not contain any inequalities, the Lagrangian dual simplifies to $D = \max_{\lambda \in \mathbb{R}^p} g(\lambda)$ or more specifically

$$D(\theta, \lambda) = \max_{\lambda} g(\lambda) = \max_{\lambda} \min_{\theta} -u_P(\pi_\theta) + \lambda \mathcal{F}_P(\pi_\theta) \tag{4.8}$$

where the optimal solution of $D(\theta, \lambda)$ is called $d*$. Gradient based approaches offer a general solution to this problem, assuming that both the utility and fairness functions are differentiable. One of the algorithms that optimizes both θ and λ at the same time is called the dual gradient method. This algorithm can be broken down into the following two optimization steps

1. $\theta_t \leftarrow \min_\theta \mathcal{L}(\theta, \lambda_t) = \min_\theta -u_P(\pi_\theta) + \lambda_t \mathcal{F}_P(\pi_\theta)$
2. $\lambda_{t+1} \leftarrow \lambda_t + \alpha \frac{\partial \mathcal{L}(\theta_t, \lambda_t)}{\partial \lambda_t} = \lambda_t + \alpha \left(\mathcal{F}_P(\pi_{\theta_t})\right)$

which are repeated until convergence is reached, with the resulting algorithm 2 minimizing the lower bound $g(\lambda)$ of the constrained optimization problem.

One of the issues with this formulation, is the fact that enforcing a strict equality might constitute an overly strong restriction of the overall solution space, which might lead to difficulties when applying any iterative, gradient based approach such as the dual gradient algorithm. In a worst case scenario, it might even lead to a solution space, where the trivial solution of rejecting every individual is the only solution that satisfies the constraint. For that reason prior works in the field of fair machine learning, such as Zafar et al. (2019), propose relaxations of the strict equality constraint for fairness. One such relaxation is replacing the equality

$$\mathcal{F}_P(\pi_\theta) = 0$$

with the two inequalities

$$\mathcal{F}_P(\pi_\theta) \geq -\delta$$
$$\mathcal{F}_P(\pi_\theta) \leq \delta$$

where δ is a constant that defines the amount of unfairness that is to be allowed within the system. Applying these changes to the constraint functions of the constrained optimization problem 4.3 the following new constrained optimization problem arises

$$\begin{aligned} \underset{\theta}{\text{argmax}} \quad & u_P(\pi_\theta) \\ s.t. \quad & \mathcal{F}_P(\pi_\theta) \geq -\delta \\ & \mathcal{F}_P(\pi_\theta) \leq \delta \end{aligned}$$

which can be brought into the standard form as defined by 4.4:

$$\begin{aligned} \underset{\theta}{\text{argmin}} \quad & -u_P(\pi_\theta) \\ s.t. \quad & -\mathcal{F}_P(\pi_\theta) - \delta \leq 0 \\ & \mathcal{F}_P(\pi_\theta) - \delta \leq 0 \end{aligned}$$

Given the new optimization problem in the standard form the Lagrangian dual D can be formulated as

$$\begin{aligned} D(\theta, \lambda) &= \underset{\nu_1, \nu_2 \in \mathbb{R}^+}{\text{argmax}}\, g(\nu_1, \nu_2) \\ &= \underset{\nu_1, \nu_2 \in \mathbb{R}^+}{\text{argmax}}\, \underset{\theta}{\text{argmin}} -u_P(\pi_\theta) + \nu_1(-\mathcal{F}_P(\pi_\theta) - \delta) + \nu_2(\mathcal{F}_P(\pi_\theta) - \delta) \end{aligned} \tag{4.9}$$

where the parameters that have to be optimized are ν_1, ν_2 and θ. This leads to the following changes in the dual gradient algorithm proposed previously:

1. $\theta_t \leftarrow \text{argmin}_\theta\, \mathcal{L}(\theta_t, \nu_1^t, \nu_2^t) = \text{argmin}_\theta\, -u_P(\pi_\theta) + \nu_1(-\mathcal{F}_P(\pi_\theta) - \delta) + \nu_2(\mathcal{F}_P(\pi_\theta) - \delta)$
2. $\nu_1^{t+1} \leftarrow \nu_1^t + \max\left[0, \alpha \frac{\partial \mathcal{L}(\theta_t, \nu_1^t, \nu_2^t)}{\partial \nu_1^t}\right] = \nu_1^t + \max\left[0, \alpha\left(-\mathcal{F}_P(\pi_{\nu_1^t}) - \delta\right)\right]$
3. $\nu_2^{t+1} \leftarrow \nu_2^t + \max\left[0, \alpha \frac{\partial \mathcal{L}(\theta_t, \nu_1^t, \nu_2^t)}{\partial \nu_2^t}\right] = \nu_2^t + \max\left[0, \alpha\left(\mathcal{F}_P(\pi_{\nu_2^t}) - \delta\right)\right]$

4.3 Augmented Lagrangian method

As discussed in the previous section, the goal of the Dual Consequential learning algorithm 2 is to solve the Lagrangian dual problem 4.7 which constitutes the lower bound of the original primal optimization problem P. Given certain assumptions and conditions, theoretical guarantees can be given, about the optimal solution of the Lagrangian

dual problem being equal to the optimal solution of the original primal problem. For example, under the assumption that both the optimization target, as well as all the constraints are convex, Slater's condition is both necessary and sufficient, meaning that any solution that satisfies Slater's condition, is a global optimum for the given optimization problem.

In the setting of fair decision making, neither the optimization target nor the constraints of the optimization problem are necessarily convex, meaning there exist no guarantees, that the local optimum found by the proposed Dual Consequential learning algorithm is an actual global optimum, meaning solutions with non-zero duality gap exist. For that reason, a large number of different publications tackle the issue finding general guarantees, about the duality gap in non-convex constrained optimization. To alleviate the shortcomings of the Lagrangian method Powell (1969) introduced the so called augmented Lagrangian which has been studied extensively over there year, for example by Rockafellar (1974), leading to new theoretical guarantees regarding the duality gap for non-convex optimization. More recently Huang & Yang (2005) produced the following result regarding the duality gap of the augmented Lagrangian method:

> "In the context of a mathematical program with both equality and inequality constraints, we proved that the second-order conditions of the Lagrangian problems with a convex quadratic augmenting function converge to that of the original constrained problem."

In our case, the augmentation function, just as all the other functions is non-convex. Nonetheless it is still worth considering the Augmented Lagrangian Method, as other works like Birgin & Martínez (2005) show empirically, that it is better behaved in terms of convergence compared to the dual gradient method.

The fundamental idea of the augmented Lagrangian, is to modify the Lagrangian function 4.5 in such a way, that the duality gap becomes zero for even non-convex constrained optimization. The most rudimentary version of this modification, which is also called the methods of multipliers, was first introduced by Powell (1969). It modifies the Lagrangian function of optimization problems constrained by equality constraints, by simply adding a quadratic penalty term of the same form as discussed in section 3.4 to the Lagrangian function, leading to the following formulation:

$$\mathcal{L}(x, \nu, \lambda) = f_0(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \frac{c}{2} \sum_{i=1}^{p} h_i(x)^2 \tag{4.10}$$

An explanation as to why this extension is reasonable, can be derived from the original optimization problem. Recall that the primal of the constrained problem can be written as the following min-max problem, as also described in equation 4.6

$$\min_x \max_\lambda f(x) + \sum_{i=1}^{p} \lambda_i h_i(x)$$

The problem with this formulation is, that "the $\max_\lambda$ function is highly non-smooth w.r.t. x."Figueiredo & Wright (2016) To smooth the function, one can add a so called proximal point term $-\frac{1}{2c}||\lambda - \bar{\lambda}||^2$ which penalizes large changes in λ from some prior estimate $\bar{\lambda}$, giving rise to the following min-max problem:

$$\min_x \max_\lambda f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) - \frac{1}{2c}||\lambda - \bar{\lambda}||^2$$

To develop a gradient based update rule to maximize λ, lets first take the derivative with regards to λ

$$\begin{aligned}\frac{\partial}{\partial \lambda_i} f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) - \frac{1}{2c}||\lambda - \bar{\lambda}||^2 &= h_i(x) - \frac{2}{2c}(\lambda_i - \bar{\lambda}_i)(1-0) \\ &= h_i(x) - \frac{1}{c}\lambda_i + \frac{1}{c}\bar{\lambda}_i\end{aligned}$$

and afterwards set the derivative to zero:

$$\begin{aligned} h_i(x) - \frac{1}{c}\lambda_i + \frac{1}{c}\bar{\lambda}_i &= 0 \\ &\Leftrightarrow \\ \frac{1}{c}\lambda_i &= h_i(x) + \frac{1}{c}\bar{\lambda}_i \\ &\Leftrightarrow \\ \lambda &= \bar{\lambda} + c h_i(x)\end{aligned}$$

Inserting λ back into the original maximization function, gives the form defined by 4.10. Given this new augmented Lagrangian function, the iterative training steps of the dual gradient method are modified as follows:

1. $\theta_t \leftarrow \min_\theta \mathcal{L}(\theta, \lambda_t) = \min_\theta -u_P(\pi_\theta) + \lambda_t \mathcal{F}_P(\pi_\theta) + \frac{c}{2}\mathcal{F}_P(\pi_\theta)^2$
2. $\lambda_{t+1} \leftarrow \lambda_t + c\,(\mathcal{F}_P(\pi_{\theta_t}))$

4.4 Empirical Results

4.4.1 Experimental Setup

The experimental setup is the same as discussed in section 3.7: The policy is trained in an online manner over 50 time steps using the same strategies $D_{\leq t}$ and D_t discussed previously. The parameters settings for the policy learning are also the same as discussed

in 3.7 and can be found in table 6.1. The setting of the new hyper-parameters introduced by the algorithms `DualConsequentialLearning` and `AugmentedDualConsequentialLearning` can be found in table 6.2.

4.4.2 Results

Generally speaking the results of the experiments on display in figure 4.4 imply, that none of the proposed dual gradient algorithm is able to find an optimal trade-off parameter λ in which some utility is maintained, while at the same time fairness is enforced. According to the experimental results, the proposed methods only manage to train a decision policy π that achieves a balance between utility and fairness for the one-dimensional FICO dataset. For both the COMPAS and the adult datasets the resulting policies are degenerate, meaning they either accept or reject everyone, leading to a solution that satisfies fairness, but does not maintain utility at the same time.

While the expectation that the use of the relaxed fairness constraint introduced in section 4.1.2 would improve the convergence of the dual gradient algorithm was reasonable, the results in figure 4.5 show, that the opposite is the case for most datasets. This behaviour is likely explained by the fact, that the connection between the covariance of decision and the benefit of difference is only established under a specific set of assumptions, laid out in section 4.1.2, mainly that the the protected group has to be the same size, as the unprotected group within the training data. If these assumptions are not fulfilled, the covariance of decision is just an approximation of the benefit of difference, without any guarantees. To understand the behaviour of the trade-off parameter λ better, figure 4.3 illustrates the behaviour of λ over time, on the COMPAS dataset, for both the covariance of decision as well as the benefit difference.

All further extensions were only tested with the benefit difference as the fairness constraint function, as the covariance of decision performed worse on the majority of the tested datasets. Allowing for a small δ of unfairness, as described at the end of section 4.2, does improve the convergence on the COMPAS dataset for D_t and reduces the variance when applied on the adult dataset, as can be seen in figure 4.6. Extending the original `DualConsequentialLearning` algorithm with a quadratic augmentation term leading to the `AugmentedDualConsequentialLearning` was a success as well, in the sense that it again slightly improves convergence for the COMPAS dataset and also reduces variance significantly for the adult dataset, as can be seen in figure 4.7.

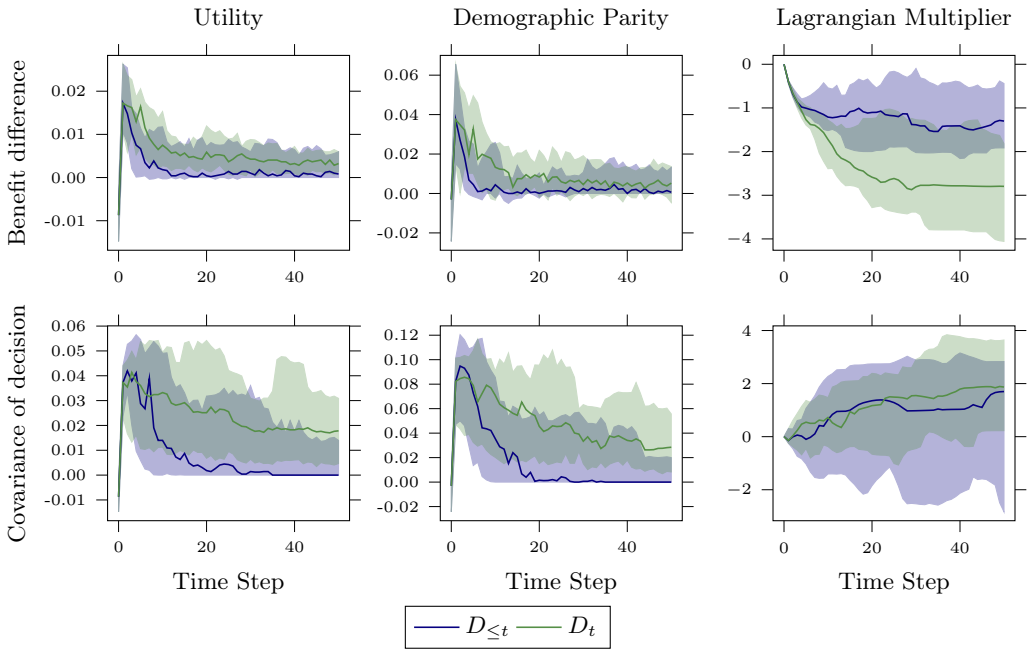

Figure 4.3: Comparing the behaviour of the benefit difference (top) with the covariance of decision (bottom) on the COMPAS dataset using the vanilla `DualConsequentialLearning` algorithm without any extensions. The top row shows the results when enforcing fairness by using the benefit of difference fairness function, while the bottom row displays the same when enforcing the covariance of decision. In both cases the median value of the lagrangian multiplier λ stabilizes after the demographic parity has gone towards zero. But just as with the utility and demographic parity, when enforcing the covariance of decisions instead of the benefit difference as the fairness constraint, the resulting lagrangian multiplier λ tends to be a lot noisier.

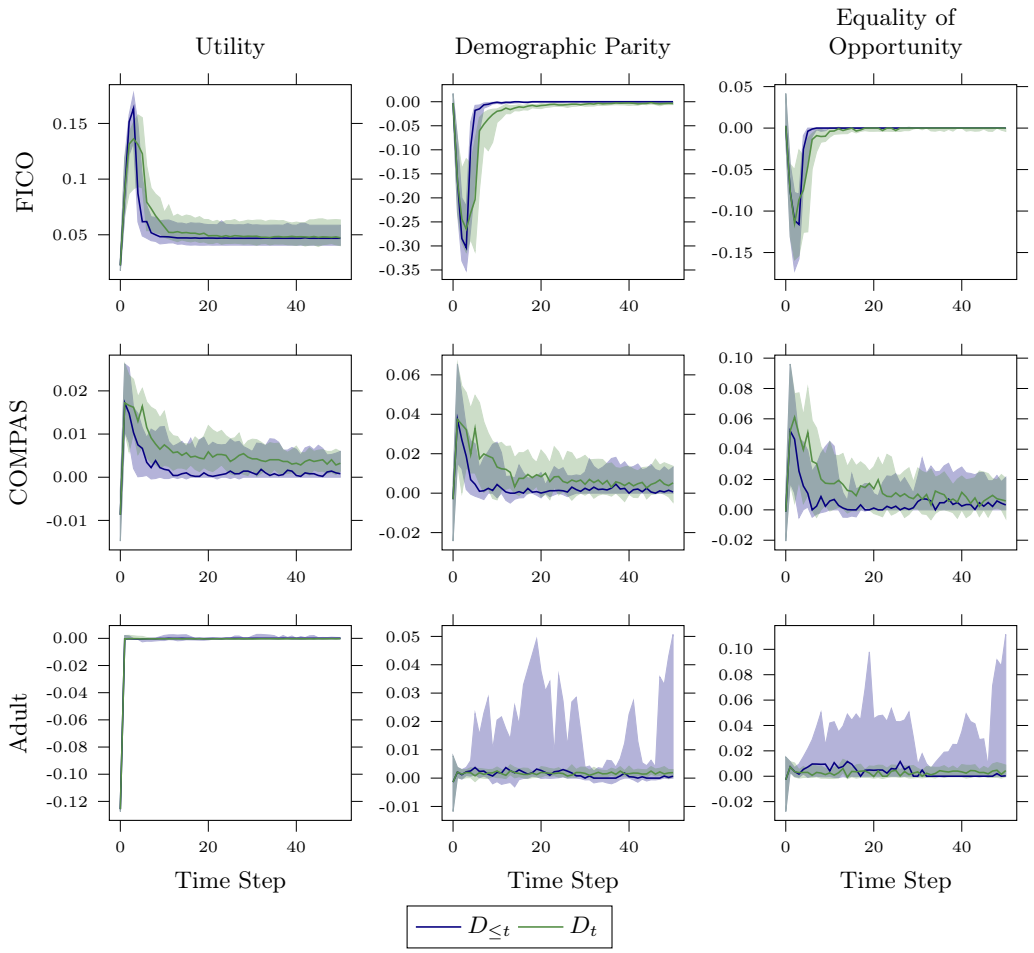

Figure 4.4: The performance of the vanilla **DualConsequentialLearning** algorithm without any extensions introduced in section 4.2 using the **benefit of difference** as the function to enforce fairness evaluated on all datasets. The top row shows performance of the algorithm evaluated on the FICO dataset, the middle row contains the results on the COMPAS dataset and the bottom row displays the results on the adult dataset. One can immediately see that the algorithm finds a local optimum for all three datasets. For the FICO data in the top row, the algorithm finds a setting of lambda, such that both utility and fairness are ensured. For the more difficult, higher-dimensional settings of the COMPAS and adult datasets, the algorithm only manages to find the trivial solution, where the policy either rejects or accepts all applicants, leading to a median utility of 0. Additionally one can see, that the results on the higher dimensional problems are much noisier, than on the FICO dataset.

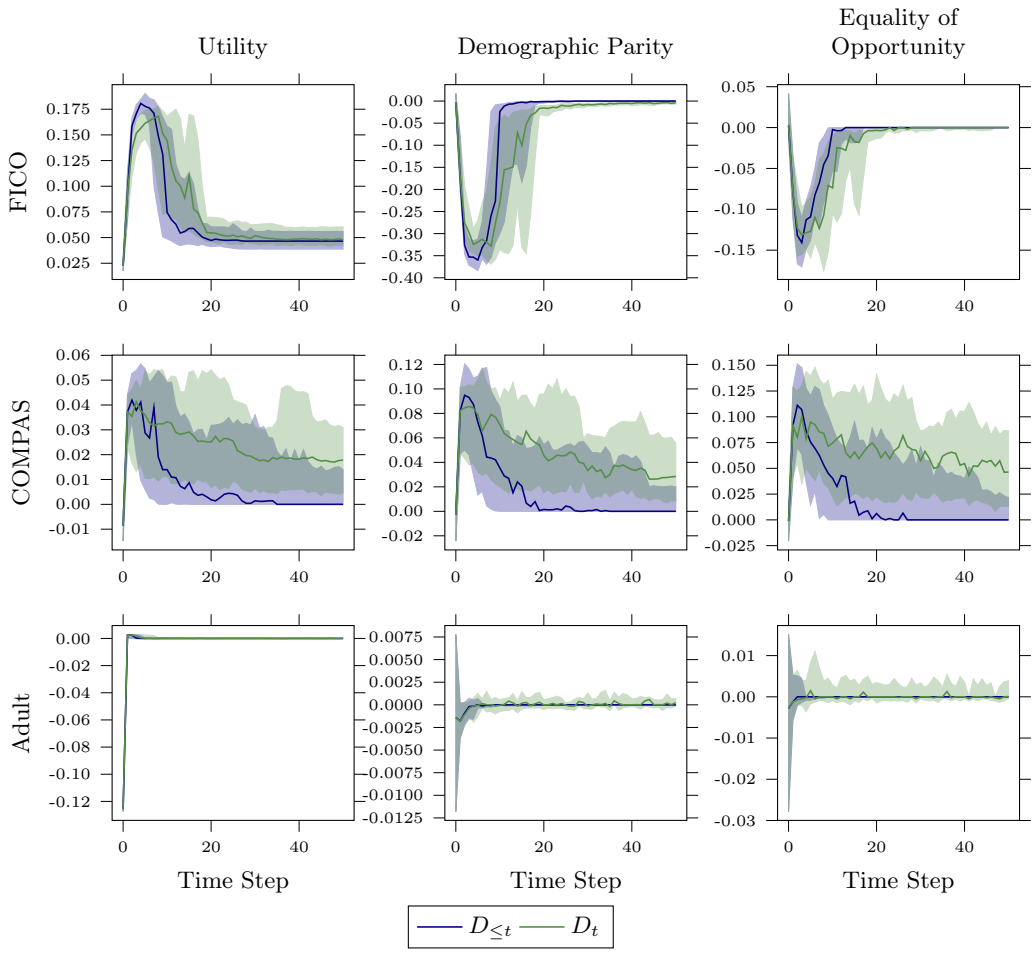

Figure 4.5: The performance of the vanilla **DualConsequentialLearning** algorithm without any extensions using the **covariance of decisions** introduced in section 4.1.2 as the function to enforce fairness evaluated on all datasets. The top row shows performance of the algorithm evaluated on the FICO dataset, the middle row contains the results on the COMPAS dataset and the bottom row displays the results on the adult dataset. Compared to the results of enforcing the benefit difference as the fairness constraint, the results on all datasets except for the adult dataset are more noisy.

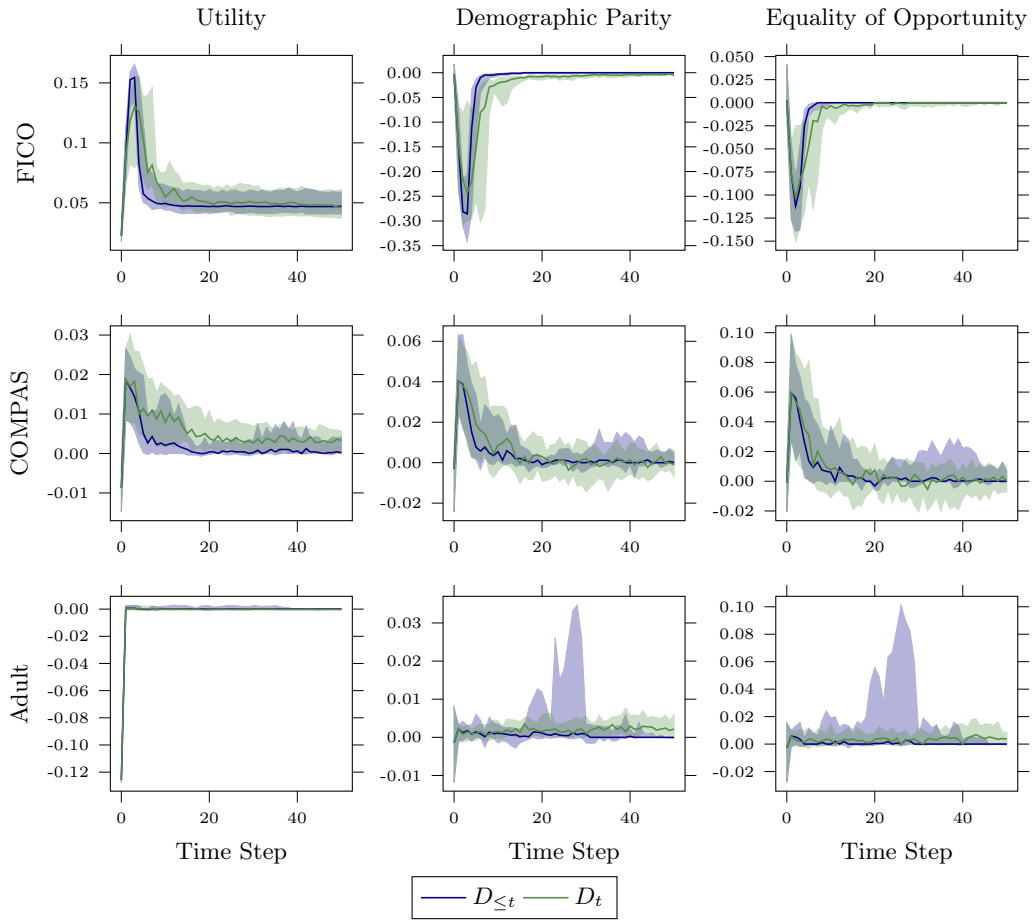

Figure 4.6: The performance of the vanilla **DualConsequentialLearning** algorithm without any extensions using the **benefit of difference** as the function to enforce fairness and allowing for a small degree of unfairness as described in section 4.2. In this case the **degree of unfairness was chosen to be** $\delta = 0.01$ and performance was again evaluated on all datasets. The top row shows performance of the algorithm evaluated on the FICO dataset, the middle row contains the results on the COMPAS dataset and the bottom row displays the results on the adult dataset. Compared to the performance of the same algorithm when enforcing the equality of $\mathcal{F}(\pi) = 0$ as shown in figure 4.4, allowing for a small δ of unfairness, seems to improve the convergence for the COMPAS dataset. Specifically the scenario where the policy is only trained on data collected in the current time step step converges to a fair state in terms of demographic parity faster. On the adult credit dataset the general noise level for both fairness measures was reduced as well.

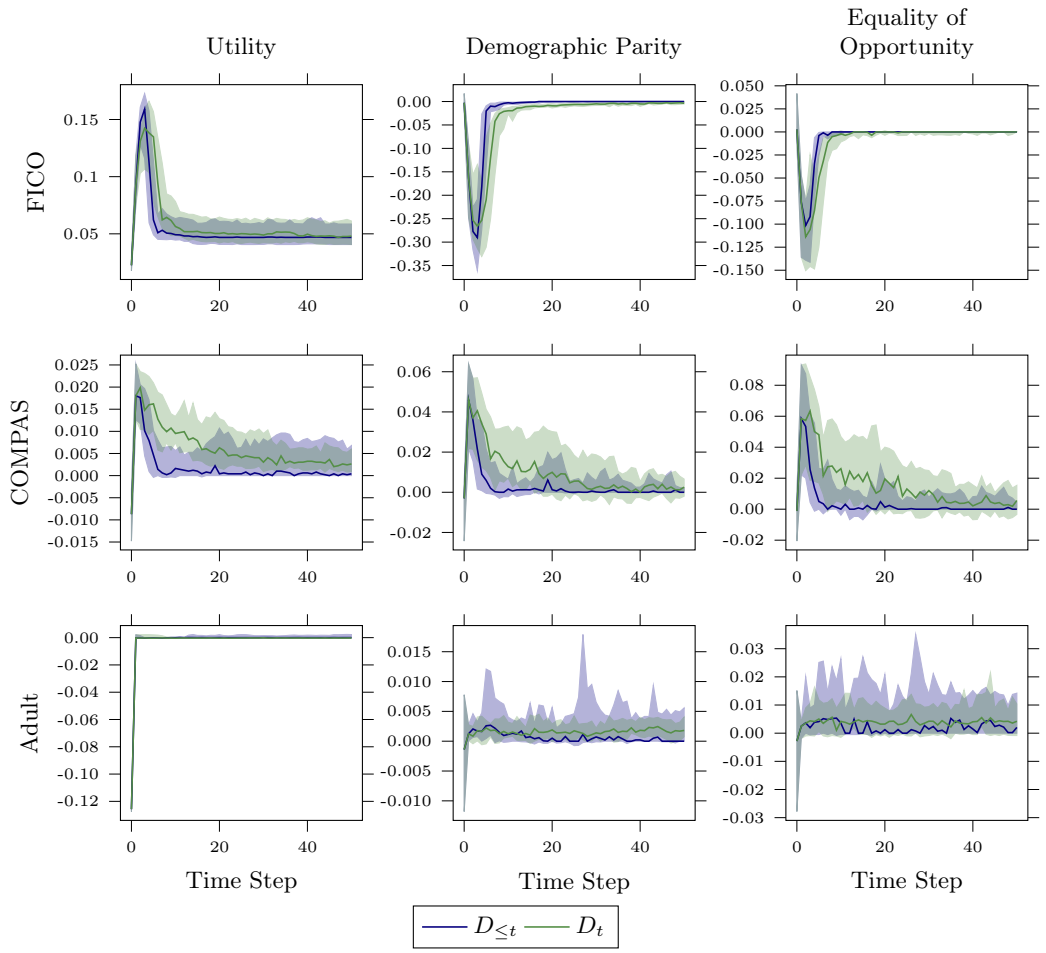

Figure 4.7: The performance of the **AugmentedDualConsequentialLearning** algorithm introduced in section 4.3 using the **benefit of difference** as the function to enforce fairness evaluated on all datasets. The top row shows performance of the algorithm evaluated on the FICO dataset, the middle row contains the results on the COMPAS dataset and the bottom row displays the results on the adult dataset. Among all the extensions to the original `DualConsequentialLearning` algorithm made, the results of the augmented version are the most promising. Convergence seem to be slightly improved for the COMPAS dataset, especially in scenario D_t where the policy is only trained on data collected in the current time step. In the case of the adult dataset the variance among the fairness measures has decreased as well.

4.5 Deep logistic regression

Kilbertus et al. (2019) propose to use logistic regression to model the decision policy π_θ with a logistic regression model. This decision was made by the authors, as the choice of a logistic regression model allows to formulate the gradient of the optimization target 3.6 analytically as seen in section 3.6. Logisitic models however are quite limited in the number of parameters and are therefore limited in their expressiveness, which might limit the possible solution space that can be modelled. Thanks to the advances of deep learning in recent years and the rapid advancement and proliferation of frameworks and libraries enabling deep learning, formulating the gradient analytically, is no longer necessary. Instead the automatic differentiation algorithms of such toolboxes as PyTorch can be used to train deep neural networks given a specified loss (or utility) function.

As one final effort to improve the performance of the methods proposed by this thesis as seen in section 4.4, the logistic decision policy proposed by Kilbertus et al. (2019), was replaced by a deep logistic regression model, using a neural network with four hidden layers consisting of 128 neurons each. As activation function between each layer the ReLU function was chosen. Figure 4.8 shows the results of running the `AugmentedDualConsequentialLearning` using DLR to model the decision policy π. Increasing the number of parameters by using a neural network did significantly reduce the variance of the results, but it also did not contradict any of the larger points discussed in section 4.4. Even though the variance in the results was largely decreased, the increased number of parameters did not change the fact, that for higher dimensional datasets like the COMPAS dataset or the adult dataset, the presented methods are unable to find a trade-off parameter λ that manages to enforce fairness, while still maintaining a degree of utility.

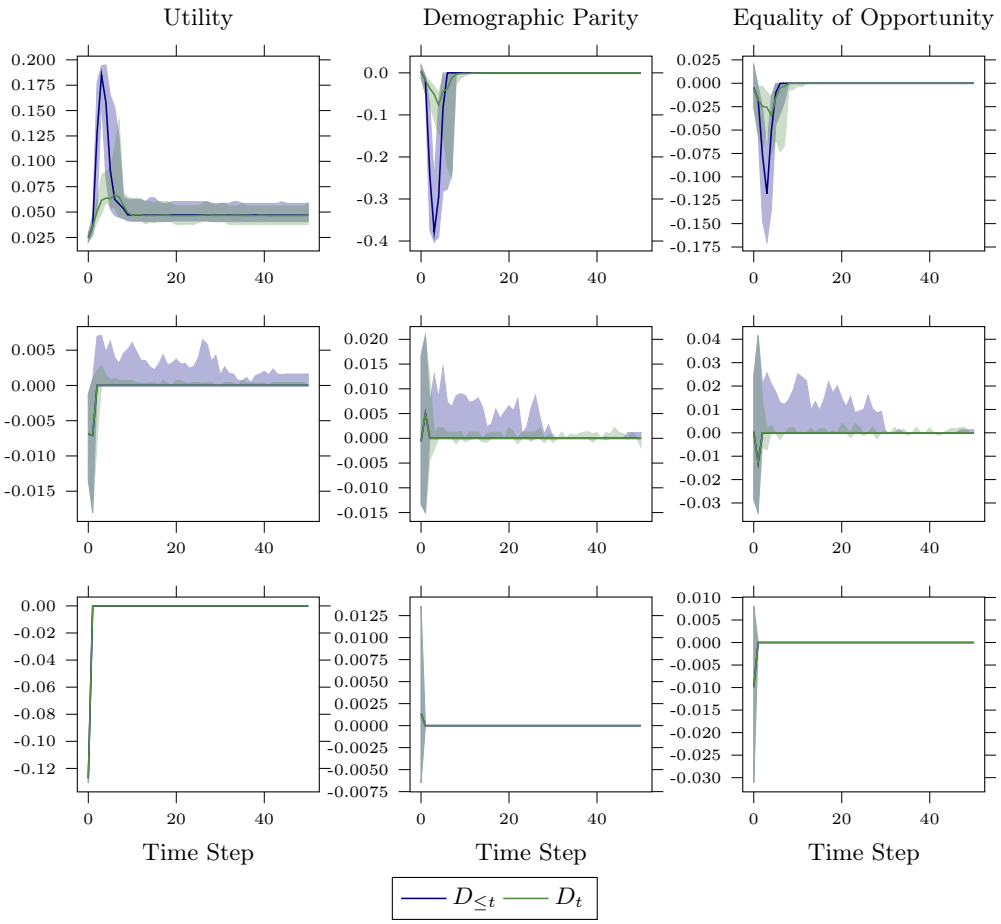

Figure 4.8: The performance of the **AugmentedDualConsequentialLearning** algorithm using the **benefit of difference** as the function to enforce fairness and **deep logistic regression** to model the decision policy evaluated on all datasets. The top row shows performance of the algorithm evaluated on the FICO dataset, the middle row contains the results on the COMPAS dataset and the bottom row displays the results on the adult dataset. Compared to the same results using the logistic policy (see figure 4.7) the variance is reduced greatly, but the overall findings remain the same: In the higher dimensional applications, no policy that enforces fairness, while at the same time preserving utility can be found.

5 Discussion and Future Work

The main conclusion that can be drawn from the results presented in this thesis, is that consequential decision making under imperfect data is a non-trivial problem, that requires a lot of additional research in the future. None of the potential solutions proposed by this thesis find a satisfying trade-off between fairness and utility within the framework of fair decision making under imperfect data as described by Kilbertus et al. (2019). Only for the simplest of all the examined real world problems the proposed algorithm was able to find a non-trivial decision policy that preserved a certain degree of utility while at the same time enforcing the fairness constraint of demographic parity. As described in section 3.7 there exists a certain λ value for which there is a steep drop in from high utility and high unfairness to no utility and no unfairness. None of the examined problems exhibit a smooth transition between between these two state, implying that this cliff might be insurmountable, by mealy extending the framework proposed by Kilbertus et al. (2019). There are multiple reasons why this might be the case:

- The utility function $u_P(\pi) = \mathbb{E}_{\substack{x,y,s\sim P(\boldsymbol{x},s,y) \\ d\sim\pi(d|\boldsymbol{x},s)}}[yd - cd]$ originally proposed by Corbett-Davies et al. (2017) might not be expressive enough. More specifically, it encodes all rejections as zero utility, as the decision maker is unaware of the results of a rejection. This does not take into account lost utility of wrongful rejections.
- As seen in the results of section 4.4 while the covariance of decisions was not the hoped for improvement of the benefit of difference, the selection of the fairness function is still a major area of concern. The empirical results show, that both fairness functions $\mathcal{F}$ explored by this thesis exhibit non-smooth, noisy behaviour when training λ using gradient based methods.
- Inverse probability weighting, even in its stabilized from as introduced in section 3.3, still causes numerical instability to the training process, which has unpredictable consequences for the stability of said process.

All of these issues imply, as stated before, that consequential decision making under imperfect data is an area of machine learning that requires additional research. Potential future works might want to reexamine both the utility function as well as the fairness function chosen to represent consequential decision making within the context of constrained optimization. Additionally it might be necessary to rethink the framework within which consequential decision making under imperfect data is viewed in its entirety. One might for example try and apply the lessons learned in the area of semi-

supervised learning, as discussed for example by Berthelot et al. (2019) and Kingma et al. (2014), to take both accepted and rejected individuals into account when training a decision making policy. Such an approach might remove the necessity of inverse probability weighting and would also increase the amount of data that is being used for training, theoretically decreasing overall noise. Any or all of these approaches should be considered for future work in this area.

Bibliography

Barocas, S., Hardt, M. & Narayanan, A. (2019), *Fairness and Machine Learning*, fairmlbook.org. `http://www.fairmlbook.org`.

Berk, R., Heidari, H., Jabbari, S., Kearns, M. & Roth, A. (2018), 'Fairness in criminal justice risk assessments', *Sociological Methods & Research* **104**(6).

Berthelot, D., Carlini, N., Goodfellow, I. J., Papernot, N., Oliver, A. & Raffel, C. (2019), 'Mixmatch: A holistic approach to semi-supervised learning', *CoRR* **abs/1905.02249**.
URL: *http://arxiv.org/abs/1905.02249*

Birgin, E. G.and Castillo, R. A. & Martínez, J. M. (2005), 'Numerical comparison of augmented lagrangian algorithms for nonconvex problems', *Computational Optimization and Applications* .
URL: *https://doi.org/10.1007/s10589-005-1066-7*

Blake, C. & Merz, C. (1998), 'UCI repository of machine learning databases', Department of Information and Computer Sciences, University of California, Irvine.
URL: *http://www.ics.uci.edu/mlearn/MLRepository.html*

Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P. & Snelson, E. (2013), 'Counterfactual reasoning and learning systems: The example of computational advertising', *Journal of Machine Learning Research* **14**(65), 3207–3260.
URL: *http://jmlr.org/papers/v14/bottou13a.html*

Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press, New York, NY, USA.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S. & Huq, A. (2017), 'Algorithmic decision making and the cost of fairness', *CoRR* **abs/1701.08230**.
URL: *http://arxiv.org/abs/1701.08230*

Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B. & Aigbavboa, C. O. (2018), A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks, *in* '2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)', pp. 92–99.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O. & Zemel, R. S. (2011), 'Fairness through

awareness', *CoRR* **abs/1104.3913**.
URL: *http://arxiv.org/abs/1104.3913*

Figueiredo, M. A. T. & Wright, S. J. (2016), 'Augmented lagrangian methods', Lecture.
URL: *https://www.him.uni-bonn.de/fileadmin/him/Section6_HIM_v1.pdf*

Hardt, M., Price, E. & Srebro, N. (2016), 'Equality of opportunity in supervised learning', *CoRR* **abs/1610.02413**.
URL: *http://arxiv.org/abs/1610.02413*

Hardt, M. & Simchowitz, M. (2018), 'Course notes for ee227c (spring 2018): Convex optimization and approximation'.
URL: *https://ee227c.github.io/notes/ee227c-lecture14.pdf*

Hernán, M. A. & Robins, J. M. (2020), *Causal Inference: What if.*, Boca Raton: Chapman & Hall/CRC.

Horvitz, D. G. & Thompson, D. J. (1952), 'A generalization of sampling without replacement from a finite universe', *Journal of the American Statistical Association* **47**(260), 663–685.
URL: *https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1952.10483446*

Huang, X. & Yang, X. (2005), 'Further study on augmented lagrangian duality theory', *Journal of Global Optimization* **31**, 193–210.

Julia Angwin, Jeff Larson, S. M. & Kirchner, L. (2016), 'Machine bias'.
URL: *https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*

Kilbertus, N., Gomez-Rodriguez, M., Schölkopf, B., Muandet, K. & Valera, I. (2019), 'Improving consequential decision making under imperfect predictions', *CoRR* **abs/1902.02979**.
URL: *http://arxiv.org/abs/1902.02979*

Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization'. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
URL: *http://arxiv.org/abs/1412.6980*

Kingma, D. P., Rezende, D. J., Mohamed, S. & Welling, M. (2014), 'Semi-supervised learning with deep generative models', *CoRR* **abs/1406.5298**.
URL: *http://arxiv.org/abs/1406.5298*

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger, eds, 'Advances in Neural Information Processing Systems 25', Curran Associates, Inc., pp. 1097–1105.
URL: *http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf*

Kusner, M. J., Loftus, J., Russell, C. & Silva, R. (2017), Counterfactual fairness, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, eds, 'Advances in Neural Information Processing Systems 30', Curran Associates, Inc., pp. 4066–4076.
URL: *http://papers.nips.cc/paper/6995-counterfactual-fairness.pdf*

Platt, J. C. & Barr, A. H. (1988), Constrained differential optimization, *in* D. Z. Anderson, ed., 'Neural Information Processing Systems', American Institute of Physics, pp. 612–621.
URL: *http://papers.nips.cc/paper/4-constrained-differential-optimization.pdf*

Powell, M. J. D. (1969), 'A method for nonlinear constraints in minimization problems', *R. Fletcher, Ed.* .

Rockafellar, R. T. (1974), 'Augmented lagrange multiplier functions and duality in nonconvex programming', *SIAM Journal on Control* .
URL: *https://epubs.siam.org/doi/10.1137/0312021*

Sagnol, G. (2017), 'Chapter vii: Duality', University Lecture.
URL: *https://www.coga.tu-berlin.de/fileadmin/i26/download/AG_DiskAlg/FG_KombOptGraphAlg/teaching/adm3_cvx_ws17/chapter7_duality.pdf*

U.S. Federal Reserve (2007), 'Report to the congress on credit scoring and its effects on the availability and affordability of credit'.
URL: *https://www.federalreserve.gov/boarddocs/rptcongress/creditscore/creditscore.pdf*

Williams, R. J. (1992), 'Simple statistical gradient-following algorithms for connectionist reinforcement learning', *Machine Learning* **8**, 229–256.

Woodworth, B. E., Gunasekar, S., Ohannessian, M. I. & Srebro, N. (2017), 'Learning non-discriminatory predictors', *CoRR* **abs/1702.06081**.
URL: *http://arxiv.org/abs/1702.06081*

Zafar, M. B., Valera, I., Gomez-Rodriguez, M. & Gummadi, K. P. (2019), 'Fairness constraints: A flexible approach for fair classification', *Journal of Machine Learning Research* **20**(75), 1–42.
URL: *http://jmlr.org/papers/v20/18-262.html*

6 Appendix

6.1 Parameter settings for empirical experiments

	FICO dataset	COMPAS dataset	Adult dataset
Training samples per time step N	128	98	781
Learning rate α	0.1	0.1	0.1
Learning rate α (DLR)	0.001	0.001	0.001
Batch size B	98	64	256
Maximum epochs E	50	50	50

Table 6.1: The parameter settings for the relevant hyper-parameters across all tested datasets for the results of the sweep over fairness constants presented in section 3.7

	FICO dataset	COMPAS dataset	Adult dataset
Learning rate α_λ	0.01	0.05	0.5
Batch size B_λ	no batching	no batching	no batching
Maximum epochs E_λ	10	10	10

Table 6.2: The parameter settings for the relevant hyper-parameters across all tested datasets for the results of the dual gradient algorithm presented in section 4.4

6.2 Consequential Learning algorithm

Algorithm 1 `ConsequentialLearning`

Require:
Cost parameter λ, number of time steps T, number of decisions N, number of epochs E, minibatch size B, learning rate α

$\boldsymbol{\theta}_0 \leftarrow$ InitializePolicy()
for $t = 0, \ldots, T-1$ **do**
  $\mathcal{D}^t \leftarrow$ CollectData($\boldsymbol{\theta}_t$, N)
  $\boldsymbol{\theta}_{t+1} \leftarrow$ UpdatePolicy($\boldsymbol{\theta}_t$, $\mathcal{D}^t$, M, B, α, λ)
  return $\{\pi_{\theta_t}\}_{t=0}^{T}$

function CollectData(θ, N)
  $\mathcal{D} \leftarrow \emptyset$
  for $i = 1, \ldots, N$ **do**
    $(\boldsymbol{x_i}, s_i) \sim P(\boldsymbol{x}, s)$
    $d_i \sim \pi_\theta(\boldsymbol{x}, s)$
    if $d_i = 1$ **then**
      $y_i \sim P(y \mid \boldsymbol{x}, s)$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{x_i}, s_i, y_i)\}$
  return $\mathcal{D}$

function UpdatePolicy($\boldsymbol{\theta'}$, $\mathcal{D}$, M, B, α, λ)
  $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta'}$
  for $e = 1, \ldots, E$ **do**
    while $\mathcal{D}^{(m)} \leftarrow$ Minibatch($\mathcal{D}$, B) **do**
      $\nabla \leftarrow 0$, $n_{(m)} \leftarrow 0$, $\boldsymbol{\theta}_{(m)} \leftarrow \boldsymbol{\theta}_{(e)}$
      for $(\boldsymbol{x}, s, y) \in \mathcal{D}^{(m)}$ **do**
        $d \leftarrow \pi_{\theta_{(m)}}(\boldsymbol{x}, s)$
        if $d = 1$ **then**
          $n_{(m)} \leftarrow n_{(m)} + 1$
          $\nabla \leftarrow \nabla + \nabla_{\boldsymbol{\theta}}\, u(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta'}})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(m)}}$
            $+\lambda \left(\mathcal{F}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta'}})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(m)}} \cdot \nabla_{\boldsymbol{\theta}} \mathcal{F}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta'}})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(m)}}\right)$
      $\boldsymbol{\theta}^{(m)} \leftarrow \boldsymbol{\theta}^{(m)} + \alpha \frac{\nabla}{n_{(m)}}$
    $\boldsymbol{\theta}^{(e+1)} \leftarrow \boldsymbol{\theta}^{(m)}$
  return $\boldsymbol{\theta}^{(M)}$

6.3 Dual Consequential Learning algorithm

Algorithm 2 `DualConsequentialLearning`

Require:
number of time steps T, number of decisions N, number of epochs for θ E_θ, minibatch size for θ B_θ, learning rate for θ α_θ, number of epochs for λ E_λ, minibatch size for λ B_λ, learning rate for λ α_λ

$\boldsymbol{\theta}_0 \leftarrow$ InitializePolicy(), $\boldsymbol{\lambda}_0 \leftarrow$ InitializeLagrangian()
for $t = 0, \ldots, T-1$ **do**
  $\mathcal{D}^t \leftarrow$ CollectData($\boldsymbol{\theta}_t$, N)
  $\boldsymbol{\theta}_{t+1}, \boldsymbol{\lambda}_{t+1} \leftarrow$ UpdateLambda($\boldsymbol{\theta}_t$, $\boldsymbol{\lambda}_t$, $\mathcal{D}^t$, E_θ, B_θ, α_θ, E_λ, B_λ, α_λ)
  return $\{\pi_{\theta_t}\}_{t=0}^{T}$

function UpdateLambda($\boldsymbol{\theta}'$, $\boldsymbol{\lambda}'$, $\mathcal{D}$, E_θ, B_θ, α_θ, E_λ, B_λ, α_λ)
  $\boldsymbol{\lambda}^{(0)} \leftarrow \boldsymbol{\lambda}'$, $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}'$
  for $e = 1, \ldots, E$ **do**
    $\theta^{(e+1)} \leftarrow$ UpdatePolicy($\boldsymbol{\theta}^{(e)}$, $\mathcal{D}$, E_θ, B_θ, α_θ, $\boldsymbol{\lambda}^{(e)}$)
    while $\mathcal{D}^{(m)} \leftarrow$ Minibatch($\mathcal{D}$, B) **do**
      $\nabla \leftarrow 0$, $n_{(m)} \leftarrow 0$, $\boldsymbol{\lambda}_{(m)} \leftarrow \boldsymbol{\lambda}_{(e)}$
      for $(\boldsymbol{x}, s, y) \in \mathcal{D}^{(m)}$ **do**
        $d \leftarrow \pi_{\theta_{(m)}}(\boldsymbol{x}, s)$
        if $d = 1$ **then**
          $n_{(m)} \leftarrow n_{(m)} + 1$
          $\nabla \leftarrow \nabla + \mathcal{F}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}'})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(e+1)}}$
      $\boldsymbol{\lambda}^{(m)} \leftarrow \boldsymbol{\lambda}^{(m)} + \alpha \frac{\nabla}{n_{(m)}}$
    $\boldsymbol{\lambda}^{(e+1)} \leftarrow \boldsymbol{\lambda}^{(m)}$
  return $\boldsymbol{\theta}^{(E)}$, $\boldsymbol{\lambda}^{(E)}$

function UpdatePolicy($\boldsymbol{\theta}'$, $\mathcal{D}$, E, B, α, λ)
  $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}'$
  for $e = 1, \ldots, E$ **do**
    while $\mathcal{D}^{(m)} \leftarrow$ Minibatch($\mathcal{D}$, B) **do**
      $\nabla \leftarrow 0$, $n_{(m)} \leftarrow 0$, $\boldsymbol{\theta}_{(m)} \leftarrow \boldsymbol{\theta}_{(e)}$
      for $(\boldsymbol{x}, s, y) \in \mathcal{D}^{(m)}$ **do**
        $d \leftarrow \pi_{\theta_{(m)}}(\boldsymbol{x}, s)$
        if $d = 1$ **then**
          $n_{(m)} \leftarrow n_{(m)} + 1$
          $\nabla \leftarrow \nabla + \nabla_{\boldsymbol{\theta}}\, u(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}'})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(m)}} + \lambda \cdot \nabla_{\boldsymbol{\theta}} \mathcal{F}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}'})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(m)}}$
      $\boldsymbol{\theta}^{(m)} \leftarrow \boldsymbol{\theta}^{(m)} + \alpha \frac{\nabla}{n_{(m)}}$
    $\boldsymbol{\theta}^{(e+1)} \leftarrow \boldsymbol{\theta}^{(m)}$
  return $\boldsymbol{\theta}^{(M)}$