Trade-Offs between Fairness, Interpretability, and Privacy in Machine Learning

by

Sushant Agarwal

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Algorithms have increasingly been deployed to make consequential decisions, and there have been many ethical questions raised about how these algorithms function. Three ethical considerations we look at in this work are fairness, interpretability, and privacy. These concerns have received a lot of attention in the research community recently, but have primarily been studied in isolation. In this work, we look at cases where we want to satisfy multiple of these properties simultaneously, and analyse how they interact. The underlying message of this work is that these requirements come at a cost, and it is necessary to make trade-offs between them. We have two theoretical results to demonstrate this. The first main result shows that there is a tension between the requirements of fairness and interpretability of classifiers. More specifically, we consider a formal framework to build simple classifiers as a means to attain interpretability, and show that each simple classifier is strictly improvable, in the sense that every simple classifier can be replaced by a more complex classifier that strictly improves both fairness and accuracy. The second main result considers the issue of compatibility between fairness and differential privacy of learning algorithms. In particular, we prove an impossibility theorem which shows that even in simple binary classification settings, one cannot design an accurate learning algorithm that is both ϵ-differentially private and fair (even approximately, according to any reasonable notion of fairness).

Acknowledgements

I would like to thank my advisor, Professor Shai Ben-David, for his guidance and support throughout. I had approached Shai and requested him to take me on as his student in my third semester at UWaterloo, after attending some of his CS485 lectures, and research talks. I really liked his approach to mathematics, especially the emphasis on understanding the intuition behind everything, while also not forgetting the importance of rigour. What particularly struck me about him was his enthusiasm for teaching, and his ability to explain complex topics in a very lucid manner (not to forget the wonderfully cute examples, thanks for introducing us to learning theory using papayas!). I found his approach to research very refreshing; it was motivated by practical, real-life problems, while at the same time, being backed by a sound theoretical understanding. I am thankful that he agreed to take me on as his student, even though I had very little 'prior knowledge' about machine learning. I believe I have learnt many valuable things about research, whilst being his student. Shai does not seem afraid of approaching problems that seem big, or important, or hard. He is also very good at identifying flaws in ideas, and asking the right questions. I think some of these traits eventually rub on to his students. Over time, I grew more confident in my ability to do research, and became less afraid of asking questions. In addition, Shai has always given me the freedom to pursue my own interests, at a pace I am comfortable with, and this work would not have been possible without that.

I am grateful to Professors Gautam Kamath and Yaoliang Yu for agreeing to read and review this thesis, and for offering valuable feedback. I would also like to thank Professor Bill Cook, for his wonderful CO759 course. That course was pivotal for me, as it motivated me to pursue more applied research. I am grateful to Professor Maura Grossman, for being a great confidant and offering helpful advice during many tough situations. I am indebted to Professor Naomi Nishimura, for accepting me to UWaterloo, and being understanding and accommodating of my changing research interests. I would also like to thank Professor Madhavan Mukund for all his invaluable help and support throughout my undergrad. I am grateful to Professor Sourav Chakraborty for encouraging me to pursue graduate studies. I am also indebted to all the wonderful teachers I had during my school years.

I am indebted to my program coordinators, Jo-Ann Hardy, and previously Marie Kahkejian, for all their encouragement and support, and for patiently dealing with the innumerable queries I had. I am immensely thankful for the wonderful support system at the university. In particular, I would like to thank Chris, Denise, Greg, Ibelemari, Joe, Loc, and Onkar in the CS department. Outside of the CS department, I thank Jessica, John, Nick, and Shelley, for all their support.

I also received great support from fellow graduate students of Shai: Nivasini, Shrinu, and Tosca. We enjoyed spending many hours discussing research problems together. Thanks to Nivasini and Tosca for the useful feedback and discussions relating to Chapter 3. Thank you to Vijay Menon, for his support throughout, and help with Chapter 4.

Finally, I thank my friends and family for their constant support and love.

Dedication

For my family.

Table of Contents

Chapter 1

Introduction

Technology has entered most aspects of our lives, with algorithms deployed to make consequential decisions such as predicting recidivism rates in released prisoners, and estimating the probability of an applicant returning a loan. Another possible scenario, and also the running example used in this thesis, is the case of university admissions. Students from around the world apply to the computer science program at the University of Waterloo. Instead of going through each profile manually, which is a pretty laborious task, let us say that the university decides to deploy an automated classifier to do the job for them. Now, because classifiers such as these are making decisions that are potentially life-altering for many people, there have been many ethical questions raised about how these algorithms function. We will look at three ethical considerations in this work: fairness, interpretability, and privacy.

1. We would like the algorithm to be *fair*, and not discriminate against an applicant just because of their membership in a minority/protected group (which could be a particular race, gender, religion, etc.). For example, women have historically been underrepresented in computer science programs, and we would ideally like our classifier to not unfairly discriminate against female applicants.

2. We would also like the classifier to be *interpretable*, what that intuitively means is that we would like to be able to understand how the classifier works and convincingly explain any decisions it might make.

3. The third concern is *privacy*. Now, because these decision making systems are typically machine learning models, and are trained on potentially sensitive data, we

would not like to inadvertently leak information about people in the training data, and would like to protect their privacy.

These concerns have received a lot of attention in the research community in the last few years. However, they have primarily been studied in isolation, that is, people have primarily looked at scenarios in which we would want an algorithm to satisfy one of these properties at a time. In this work, we look at cases where we want to satisfy multiple of these properties simultaneously, and analyse how these properties interact. Overall, we find that that these properties are often at odds with each other, and it is necessary to make trade-offs between them.

1.1 Contributions

We show two theoretical results to demonstrate the necessity to make trade-offs between the properties of fairness, interpretability, and privacy. The first result looks at cases where we would like to have accurate classifiers that are also fair and interpretable, and shows how the desiderata of fairness and accuracy are at odds with the property of interpretability. In the second result, we look at cases where we would like a learning algorithm to be both fair and private (while maintaining accuracy), and show that these two properties are incompatible. We elaborate on both of the results below.

1. As decision making algorithms (i.e., black box models such as deep neural networks, etc.) have become increasingly complex, they are becoming less transparent and harder to audit. This is undesirable, and we would prefer if the models were more interpretable (i.e., we are able explain their decisions).

   Creating models that are intuitively 'simple' to humans is a natural strategy to increase their interpretablilty. An example of such a simple model is linear classifiers. Another way to build simple classifiers is to reduce the number of features that are involved in the decision making process. We consider a formal framework developed in Kleinberg et al. [16] to model he construction of simple classifiers, which captures some commonly used methods of building interpretable models. We discuss the interaction between the desiderata of simplicity, fairness and accuracy of binary classifiers in this framework.

   Given a set of features, we have an optimal classifier (i.e., the most accurate classifier that can be built from the given features). One may naturally wish to simplify the

optimal classifier to increase interpretability. Another reason to simplify classifiers could be to even increase fairness in some cases. Simpler and more interpretable models can be easier to audit, and we can possibly identify sources of unfairness and correct them with more ease in simple models. Deleting features that can be potentially viewed as unfair, has also been adopted in practice, for example, in the well known "ban the box" scenario, where the check box that asks if applicants have a criminal record from hiring applications is removed [3].

In contrast, this work discusses the negative effects of building simple classifiers on their fairness. More specifically, we show that every simple classifier can be improved; i.e., replaced by a more complex classifier that strictly increases both fairness and accuracy with respect to the simple classifier. It is quite expected that using a simple model would result in a loss in accuracy, because imposing simplicity requirements on a classifier reduces its expressive power. The surprising finding here is that simplification leads to a loss in fairness as well, i.e., we can always find a more complex classifier that is more fair, in fact, we can always find a more complex classifier that is simultaneously more fair and accurate than the simple classifier. Hence, we see that that the properties of fairness and accuracy clash with interpretability (or simplicity).

2. Our second main result talks about the clash between the requirements of differential privacy and fairness in learning algorithms. Although there are many settings where one might only care about one of these issues, they are not always mutually exclusive, for one can easily think of several scenarios where one might not only need privacy but also need to ensure that the procedure is fair. A typical example of such a setting is allocation of scarce resources—be it research funding, natural resources, loans, etc. Given this, it is imperative that the issues of privacy and fairness be studied together. However, unfortunately, there has been very little work that has looked at these issues simultaneously. Especially in light of the fact that the 2020 U.S. census is going to employ differential privacy and that the annual distribution of at least 675 billion dollars relies on census data [14], we believe that having a good understanding of the privacy-fairness trade-offs involved is of prime importance.

   Towards this end, our second main result in this work is an impossibility theorem which states that even in a very simple binary classification setting, no learning algorithm that is ϵ-differentially private (for any $\epsilon \geq 0$), and approximately fair (i.e., the algorithm is guaranteed to output an approximately fair classifier) can have non-trivial accuracy. This shows how in certain applications it might be necessary to make trade-offs between privacy and fairness and how one may not be able to hope

for all these properties to hold together.

Organisation

The rest of this document is organised as follows. In Section 2 we go over some related work, especially two papers that most relate to our work. Following this, Section 3.1 introduces the framework used to model simple classifiers, and some key definitions that we will need throughout. In Section 3.2 we present our first main result, on the clash between interpretability and fairness. Section 4 talks about our result highlighting the incompatibility of privacy and fairness. Section 4.3 talks about some other directions we pursued along the trade-offs between privacy and fairness, and finally in Section 5 we conclude and present some potential directions for future work.

Chapter 2

Related Work

Although the ethical issues concerning algorithms that we discuss in this work have been considered widely in the now ubiquitous literature on differential privacy, model interpretability, and algorithmic fairness, they have mostly been considered in isolation. In particular, the literature on algorithmic fairness discusses how to handle issues such as bias and discrimination (e.g., [5, 17, 11]), the literature on model interpretability addresses the growing need for transparent models (e.g., [4, 24, 19]), and the literature on differential privacy talks about protecting the privacy of individuals (e.g., [6, 8]).

Not much previous work has looked at cases where one would want to satisfy multiple of these properties simultaneously, or analysed how these properties interact. Previously, Doshi-Valez et al. [4] argued that increasing a model's interpretability makes the model easier to analyse, and therefore assists in (a) deciding whether the model is fair and (b) modifying the model to ensure that it is. In contrast to their work, we present a result that captures the fact that the interpretability of a model could be at odds with its fairness.

Our first main result has a similar message to the work of Kleinberg et al. [16]. We try and address some of the limitations of the framework in Kleinberg et al. and prove similar results to theirs, but in what be believe to be a less restrictive framework. We will compare our work to Kleinberg et al.'s in greater detail in Section 3.1.7, and as we go along the write-up, while describing the framework and necessary terminology.

The work that is most relevant to our second main result is that of Cummings et al. [2]. Cummings et al. [2] consider the trade-offs when considering learning algorithms that satisfy differential privacy and one particular notion of fairness, namely equal opportunity (see Section 4.1 for a definition), and one of the results they claim is a weaker version of the one we have here. In particular, they claim that there is no learning algorithm that achieves

ϵ-differential privacy, satisfies equal opportunity (is guaranteed to output a classifier that satisfies equal opportunity), and has accuracy better than a constant classifier. However, to the best of our understanding, we believe that there is a gap in their argument (see Section 4.2 where we briefly describe what it is), and so unfortunately their proof idea does not go through. So, in essence, our contribution here can be summarized as correcting their proof and also generalizing it, by showing that such an impossibility holds with respect to every reasonable notion of (even approximate) fairness.

Apart from the work of Cummings et al. [2], another important paper that was a motivation for pursuing this line of work was that of Kuppam et al. [18]. Kuppam et al. [18] empirically show how there might be privacy-fairness trade-offs involved in certain settings. In particular, they consider three resource allocation settings and use census data to which noise has been added to demonstrate how adding noise so as to achieve differential privacy could disproportionately affect some groups over others in the settings that they consider.

Besides the paper mentioned above, there is also work by Dwork and Mulligan [7], Ekstrand et al. [9], and Jagielski et al. [15], where they consider the issues of privacy and fairness together. The former two mainly raise questions along the direction of tradeoffs involved, while the latter paper shows two algorithms that satisfy (ϵ, δ)-differential privacy and equalized odds. Although at first glance it may seem like these algorithms contradict the impossibility result in this work, it is important to note that it doesn't, for we are considering $(\epsilon, 0)$-differential privacy throughout.

Chapter 3

Trade-Offs between Fairness and Interpretability

Before we dive into the exact setup, let's briefly go over the broad message of this section again. A common method to build interpretable classifiers is to essentially avoid using very "complex" models such as neural networks which usually have a ton of parameters. Instead, one could choose to use models that are intuitively 'simple' for humans to understand, such as linear classifiers. We consider a formal framework that captures some commonly used methods used in the construction of simple classifiers. Then, we show that if we try to restrict our classifier to be simple within this framework, it can be replaced by a more complex classifier that strictly improves both fairness and accuracy (there are some variations of this theorem based based on different underlying assumptions, but the message of all of the statements is roughly the same). Hence, we see that, in a sense, interpretability, or simplicity in this case, is at odds with fairness and accuracy.

3.1 Formalising the Framework

Domain Set We denote the domain set by X.

Ground Truth Function Given some scenario, we assume the existence of a 'Ground Truth' function, that assigns a label to each point in the domain set, that is,

$$G^* : X \to \{0, 1\}.$$

For example, in the university admissions case, a point in the domain set is assigned the label 1 if they would do well in university, and 0 otherwise. In general, we refer to an instance labeled 1 as 'good', and 'bad' otherwise.

Remark. Our results and proofs also go through for the case where the ground truth function G^* is non-deterministic, that is, instead of being labeled 0 or 1, a particular instance might be labeled 0 with probability 0.6, and 1 with probability 0.4. However, for simplicity, we assume that the ground truth function is deterministic.

Probability Distribution We have an underlying distribution $\mathcal{D}$ over X.

Remark. If we allow G^* to be non-deterministic, the underlying distribution $\mathcal{D}$ would be over $X \times \{0, 1\}$, not X.

Features Each instance in X is represented by the set of features $F = \{f_1, \ldots, f_k\}$. For example, in the university admissions case, the features could be things like age, SAT score, school grades, and so on. Each $f_i : X \to \{0, 1\}$ is a binary feature.

Remark. For simplicity, we assume that each feature is binary. However, the results, and pretty much the same proofs also hold for the case when each feature can take finitely many values.

Protected Group Membership Each instance also belongs to one of two groups - A or D. A stands for the advantaged group, whereas D stands for the disadvantaged group. D can be thought of as the minority group that we wish to protect from discrimination. For example, in the university admissions case, the advantaged group could be thought of as males, whereas the disadvantaged group is females. The group membership feature $f_m : X \to \{A, D\}$ maps an instance to their group.

Task So what is the task at hand? Given an unlabeled set of applicants T, we want to admit a fixed fraction r (known as admission rate) of them such that we are as accurate as possible (i.e., admit as many good applicants as possible). In practice, to build this classifier, we have access to a labeled sample of points generated by the same distribution, or the training set. In our setup, we assume we have full access to the distribution and ground truth function to build the classifier.

3.1.1 Partitions and Cells

We can partition the domain set X into different parts, and we call each part a cell.

Measure Let $\mu(C)$ denote the mass of the probability distribution $\mathcal{D}$ in cell C. We will refer to this as the *measure* of C.

Some Natural Partitions

One could create a partition at random, but a more natural way to create cells is based on their feature vectors. That is, two instances are part of the same cell if and only if they have the same feature vector representation. Recall, a feature set is a set of all the attributes that one is interested in. A feature vector is a vector that stores the values of these features for a particular instance, in a specific order. For example, let's say our feature set consists of height and age. If Alice is 26 years old and she is $5'$ tall, her feature vector would be $[26, 5]$ or $[5, 26]$ depending on our choice of how to order the elements. The order needs to be consistent across different entries.

Recall, we are given access to a set of features $F = \{f_1, \ldots, f_k\}$. We also had the group membership feature f_m and if we append that to the feature set F, we denote the resultant feature set by F'. The partition induced by F is denoted by f, and we denote the cells of f by $C_1, \ldots, C_n$, (where $n = 2^k$, because each feature is binary). The partition induced by F' is denoted by f', and consists of 2^{k+1} cells, as there are $k+1$ binary features. The cells in f' are obtained by splitting each cell in f into two parts, according to the group membership feature f_m. For e.g., C_1 is split into C_1^A and C_1^D, which represent the advantaged and disadvantaged people in the cell C_1 respectively. We denote the cells of f' by $C'_1, \ldots, C'_{2n}$.

3.1.2 Score Function

Score of a Cell

We say that the probability of a random instance sampled according to $\mathcal{D}$ being good (given that it lies in some cell C) is the "score" of C. We denoted the score of C by $S(C)$, i.e.,

$$S(C) = \Pr_{x \sim \mathcal{D}}[G^*(x) = 1 \mid x \in C]$$

Score of an Instance

By score of an instance $x \in X$, we mean the score of the cell it belongs to in the partition f'. Given the feature set, the score of an instance is the most accurate estimate we have of the probability of the instance being good.

3.1.3 Classifiers

A classifier assigns every point in the domain set a label from $\{0, 1\}$. Because each point in the domain set is represented by its feature vector, the classifier is essentially a function from the space of all feature vectors to the label set. A classifier is a function from $\{0, 1\}^{k+1} \to \{0, 1\}$ (and we allow for randomisation).

Equivalence between Classifiers and Partitions

A given partition h of the domain set and admission rate r induces a threshold classifier that we denote by h_r. We now explain how this classifier works. The classifier h_r sorts the cells of h in descending order of their scores. We then admit applicants in this order until we admit the desired fraction r.

Remark. We will often use the term classifier and partition interchangeably.

More formally, consider an arbitrary partition h which partitions X into the cells $C_1^\wedge, C_2^\wedge, \ldots C_t^\wedge, \ldots, C_d^\wedge$. We sort the cells of h in descending order of their scores. Without loss of generality assume that h partitions X into cells $C_1^\wedge, C_2^\wedge, \ldots, C_d^\wedge$ with decreasing (not necessarily strict) order of scores. We merge cells with the same scores to form a new partition h^* with cells $C_1^*, C_2^*, \ldots, C_{d'}^*$, in strictly decreasing order of scores. Now, start admitting applicants in order as follows until you admit a fraction r of them. Let r_j be the fraction of the first j cells of h^* in the order they are represented (In fact, whenever we will write a partition in this paper, we will assume the cells are ordered in descending order of their scores.). If $j(r)$ is the unique index j such that $r_{j-1} \leq r < r_j$, then the instances admitted consist of all the applicants in the cells $C_1^*, C_2^*, \ldots, C_{j(r)-1}^*$, together with a subset of $C_{j(r)}^*$ of fraction $(r - r_{j-1})$. The instances in $C_{j(r)}^*$ to be admitted will be picked randomly.

Optimal Classifier

Recall that we had discussed the partition f' above, which is the partition induced by all the features we have. Given the feature set we have, the most accurate classifier we can construct is the one induced by the partition f'.

In Practice

In reality, to build this classifier, we will only have access to a labelled sample of points generated by $\mathcal{D}$. We need to estimate the true score function values of each cell from their empirical estimates from the sample. In our setup, we assume we have full access to the distribution $\mathcal{D}$ and ground truth function G^* to build the classifier. We do this because we want to look at the behavior of these classifiers in isolation without considering any added complications due to sampling error issues.

3.1.4 Modeling Simple Classifiers

We use the framework introduced in Kleinberg et al. [16] to model the construction of simple classifiers. Two particular approaches to build simple classifiers that this framework captures are (i) shallow decision trees, and (ii) using a small number of informative features (feature selection). These approaches follow a common principle: they simplify the underlying model by combining distinguishable applicants (applicants with different feature vector representations) into larger sets and making a common decision at the level of each set. What that means in our framework, is that we would simplify f' (the most accurate classifier induced by all the features we have) (or simplify any classifier/partition in general) by combining multiple cells of it to make one cell. We define it formally below.

Some Terminology

Definition 1 (Refinement)**.** A partition r of a set X is a refinement of a partition c of X if every cell of r is a subset of some cell of c.

Definition 2 (Coarsening)**.** The partition c as in the above definition is a coarsening of r.

Definition 3 (Simplification)**.** A 'simplification' h of a partition q is a coarsening of q such that $h \neq q$.

Definition 4 (Complexification)**.** A 'complexification' h of a partition q is a refinement of q such that $h \neq q$.

Some Particular Simplification Methods

The approaches we aim to capture, which are (i) shallow decision trees, and (ii) feature selection, do not combine cells at random, but they do it in a structured way. For example, we observed that f is the simplification of f' associated with deleting the group membership feature f_m.

Below, we list some specific forms of simplification that we consider, that have some natural structure to them.

Remark. Note that deleting a feature is a specific form of simplification that halves the number of cells.

Definition 5 (Non-trivial partition)**.** A partition h is non-trivial if it contains a cell C that contains 2 instances x and y with different scores.

Definition 6 (Non-trivial cell)**.** We say that such a cell C as above is a non-trivial cell.

Definition 7 (Group Agnostic-simplification)**.** A simplication of f' such that instances differing only in the group membership feature are mapped to the same cell. This basically means that as a simplification step, the classifier is constrained at the very least to completely ignore the group membership feature. There may or may not be further simplifications on top of this.

Now we move on from group-agnostic simplifications to a more general notion of simplification called graded simplifications.

Definition 8 (Graded-simplification)**.** Consider cell partition f' of X : $C'_1, C'_2, \ldots, C'_{2n}$. Consider simplification w of f' that partitions X into the cells $C_1^\wedge, C_2^\wedge, \ldots, C_t^\wedge, \ldots, C_d^\wedge$ with descending order of scores. Each cell $C_i^\wedge \in w$ can be written as $C_i^\wedge = \cup_{j=1}^{k} C'_{i_j}$ (i.e., the union of some cells $C'_{i_1}, C'_{i_2}, \ldots, C'_{i_k} \in f'$). We denote the set of indices $\{i_1, i_2 ... i_k\}$ corresponding to $C_i^\wedge$ as $V(C_i^\wedge)$.

C^A denotes the instances of cell C that are advantaged. Similarly, C^D denotes the disadvantaged instances of cell C. A graded simplification w of f' is one where each cell $C_i^\wedge \in w$ has the property that either $V(C_i^{\wedge^A}) \subseteq V(C_i^{\wedge^D})$ or $V(C_i^{\wedge^D}) \subseteq V(C_i^{\wedge^A})$.

3.1.5 Quantifying Niceness of Classifiers

Fairness

Our fairness objective function penalises FP_A (False Positives for group A) and FN_D (False Negatives for group D), and aims to minimise a weighted sum of the two.

$$FP_A(h_r) = \text{Expected fraction of bad instances in } A \text{ that } h_r \text{ accepts}$$
$$FN_D(h_r) = \text{Expected fraction of good instances in } D \text{ that } h_r \text{ rejects}$$

For some $0 < \gamma < 1$,

$$Fairness(h_r) = \mathbb{E}[-(\gamma(FN_D(h_r)) + (1-\gamma)FP_A(h_r))]$$

Accuracy

$$Accuracy(h_r) = \frac{\text{Expected fraction of good instances } h_r \text{ accepts}}{\text{Total fraction of instances } h_r \text{ accepts (i.e., } r)}$$

Equity

$$Equity(h_r) = \frac{\text{Expected fraction of disadvantaged instances } h_r \text{ accepts}}{\text{Total fraction of instances } h_r \text{ accepts (i.e., } r)}$$

Comparing Two Classifiers Consider two partitions of X, say h and g. We say that a partition h strictly improves on partition g in criteria Q (e.g., accuracy) if for every $r \in [0,1]$, $Q(h_r)$ is at least $Q(g_r)$, and there exists an $r' \in [0,1]$ such that $Q(h_{r'})$ is strictly more than $Q(g_{r'})$.

3.1.6 Assumptions in our Framework

We state below some 'niceness' assumptions on the data, which are also made by Kleinberg et al. [16]. One of our contributions is to prove a similar result as theirs without the disadvantage assumption (stated at the end of this section), which we believe to be quite restrictive.

1. *Equality assumption:* For every cell $C_i \in f$, if we split it by group membership, both resultant cells C_i^A and C_i^D have the same score.

   This is a reasonable assumption and intuitively means that if we have enough informative features about a person, their membership in a protected group does not affect their performance.

2. *Denseness assumption:* For every cell $C_i \in f$, if we split it by group membership, both resultant cells C_i^A and C_i^D have positive measure.

3. For a set of cells $R \subseteq f'$, use $S(R)$ to denote the weighted average value of S in the cells of R.

   Genericity assumption: Let $R, T \subseteq f'$ be two distinct sets of cells such that if $R = C_i^A$ then $T \neq C_i^D$. We then assume that $S(R) \neq S(T)$.

   Remark. This in particular implies that the cells of f can be arranged in strictly descending order of scores. Without loss of generality, we assume that $S(C_1) > S(C_2) > \cdots > S(C_n)$.

The assumption below is used by Kleinberg et al. [16], but we do not use it for our results.

Disadvantage assumption: Given cells $C_i, C_j \in f$ such that $S(C_i) < S(C_j)$, then

$$\frac{\mu(C_i^A)}{\mu(C_i^D)} < \frac{\mu(C_j^A)}{\mu(C_j^D)}.$$

This condition intuitively means that for every two feature vectors a and b such that instances having feature vector representation a have a higher chance of success than instances having feature vector representation b, instances having feature vector representation a have a higher chance of belonging to the advantaged group than instances having feature vector representation b.

3.1.7 Differences with Respect to Previous Work

As mentioned before, our results are similar in spirit to the main statement of Kleinberg et al. [16]. However, our setup has some key differences and enjoys multiple advantages.

- We do not use the disadvantage assumption, which was quite a restrictive assumption.
- They use the notion of equity to quantify fairness, which essentially involves maximizing the number of minority group applicants the classifier labels positively. The notion we use to quantify fairness is more aligned with accuracy, and penalises the false negatives of the minority group, and false positives of the majority group. We believe that a desirable property of any notion of fairness is that a classifier that is perfectly accurate is also perfectly fair, which is something our notion satisfies but theirs does not.

3.2 Result

We have introduced the necessary terminology to formally state the main result of this section. We have multiple versions of the result, and based on different underlying assumptions, are able to prove slightly different things in each of them. However, the message of all of the statements is roughly the same. We show that if we try to restrict our classifier to be simple within the framework previously discussed, it can be replaced by a more complex classifier that strictly improves both fairness and accuracy. Therefore, we see that simplicity clashes with the properties of fairness and accuracy.

3.2.1 Group Agnostic Simplifications

We first consider the case where we restrict simplifications to group agnostic ones.

Result

We first informally explain the result of this section. Recall that the classifier resulting from partition f' is the most accurate classifier we can build with the features we have. If we choose to use a simpler classifier than f', say w, it might lead to an increase in fairness, interpretability, or equity, but we lose accuracy. That might have been a good trade-off, but we show that the simple classifier w is not optimal if we ignore the requirement of interpretability, as there exists a partition h (achievable by the features we have) that is simultaneously more fair, accurate, and equitable than w. Therefore, we would strictly prefer h over w, if we ignore interpretability requirements, and therefore we see that interpretability clashes with the desiderata of fairness, accuracy, and equity.

Remark. If we do not require partition h to be achievable with the features we have, and the features of an instance do not completely determine its label (which is often the case in practice), it is trivial to find an h that strictly improves in fairness and accuracy over any w (where w is a non-trivial simplification of f'). For example, the following partition would work: $h = C_1^\wedge, C_2^\wedge, C_3^\wedge, C_4^\wedge$, where $C_1^\wedge$ = Good instances in D, $C_2^\wedge$ = Good instances in A, $C_3^\wedge$ = Bad instances in D, $C_4^\wedge$ = Bad instances in A. Here we ensure to not merge any cells in h while admitting instances.

Theorem 1. *For every non-trivial group-agnostic simplification, say w, there exists a partition h (achievable by the features we have) which strictly improves accuracy, fairness and equity with respect to w.*

Proof. Consider non-trivial simplification w of f. It partitions X into the cells

$$C_1^\wedge, C_2^\wedge, \cdots, , C_j^\wedge, C_{j+1}^\wedge, \cdots C_t^\wedge, \ldots, C_d^\wedge$$

with descending order of scores. Take a non trivial cell of w, say $C_t^\wedge$. The non trivial cell $C_t^\wedge$ consists of two or more cells of f with different scores. Say $C_t^\wedge$ is the union of $C_a, C_b, \ldots, C_z \in f$. Let the cell of f in $C_t^\wedge$ with the highest score be C_b.

Construct h as follows: Remove $\epsilon > 0$ measure of X from C_b^D to create a separate cell C'. This is the new partition h. Denote the remainder of $C_t^\wedge$ by C''. Observe that $S(C') > S(C_t^\wedge) > S(C'')$. Take ϵ small enough to not change order of C'' in the partition w. It should be in the same position as $C_t^\wedge$ was before. (we can do this because of the genericity assumption) The only change in the order is that C' jumps to some position ahead of C''. The new partition h is

$$C_1^\wedge, C_2^\wedge, \cdots, C_j^\wedge, C', C_{j+1}^\wedge, \cdots C_{t-1}^\wedge, C'', C_{t+1}^\wedge \cdots, C_d^\wedge$$

with descending order of scores.

Remark. Removing $\epsilon > 0$ measure of a cell to create a separate new cell can be viewed as randomising over instances in that cell. Each instance goes to the new cell with probability ϵ, and stays in the old cell with probability $1 - \epsilon$.

We can show that for all rates r, the fairness, equity and accuracy of h is at least as good as w, and for at least one value of r, strictly better in all 3 criteria.

Case 1: $r \geq r_t$ or $r \leq r_j$:

We note that in h, the measure of all cells uptil C'' is r_t. The classifiers resulting from w and h with admission rate r as above classifies all cells the same way. Therefore, h_r has the same accuracy, equity and fairness as w_r.

Case 2: $r_j + \mu(C') \geq r > r_j$:

Both h_r and w_r classify all instances of $C_1^\wedge, \ldots C_j^\wedge$ as 1. The admission rule h_r classifies instances of $C_{j+1}^\wedge, \cdots C_{t-1}^\wedge$ as 0 and some mass $\mu = r - r_j$ of C' as 1, while the admission rule w_r classifies some mass μ of $C_{j+1}^\wedge, \cdots C_t^\wedge$ as 1, and the remaining as 0 (we start by classifying instances from from $C_{j+1}^\wedge$ as 1, if $\mu(C_{j+1}^\wedge) < \mu$, then we move on to C_{j+2}, and so on). Since the score of C' is greater than the score of each cell $C_{j+1}^\wedge, \cdots C_t^\wedge$, the mass μ of C' that h_r classifies as 1 has a higher measure of expected true 1's than the mass μ of $C_{j+1}^\wedge, \cdots C_t^\wedge$ that w_r classifies as 1. Therefore, h_r is in expectation more accurate than w_r.

The mass μ of C' that h_r classifies as 1 has a higher measure of disadvantaged instances than the mass μ of $C^{\wedge}_{j+1}, \cdots C^{\wedge}_t$ that w_r classifies as 1 because C' only consists of disadvantaged instances, while each cell in $C^{\wedge}_{j+1}, \cdots C^{\wedge}_t$ consists of both disadvantaged and advantaged instances (because of the denseness assumption). Hence, h_r has higher equity than w_r.

It is also easy to see that the mass μ of C' that h_r classifies as 1 has on expectation lower FP_A and FN_D values than the mass μ of $C^{\wedge}_{j+1}, \cdots C^{\wedge}_t$ that w_r classifies as 1. Hence, h_r has higher fairness than w_r.

Case 3: $r_t > r \geq r_j + \mu(C')$:

Both h_r and w_r classify all instances of $C^{\wedge}_1, \ldots C^{\wedge}_j$ as 1 and all instances of $C^{\wedge}_{t+1}, \ldots C^{\wedge}_d$ as 0. h_r classifies all instances of C' as 1, while w_r classifies some mass μ of them as 0 and instead classifies some mass μ from $C^{\wedge}_{j+1}, \ldots C''$ with expected score lower than that of C' as 1. This is where the two classifiers differ. Cells $C^{\wedge}_{j+1}, \ldots C''$ have a lower score and lesser proportion of disadvantaged instances than C'. Reasoning similarly as Case 2, we observe that the classifier w_r is less fair, less accurate and has lower equity than h_r. □

3.2.2 Graded Simplifications

Now we move on from group-agnostic simplifications to a more general notion of simplification, called graded simplification.

Result

We first informally explain the result of this section. If we use a simpler classifier than f', say w, it might lead to an increase in fairness, interpretability, or equity, but we lose accuracy. We show that the simple classifier w is not optimal if we ignore the requirement of interpretability, as there exists a partition h (achievable by the features we have) that is simultaneously both more fair and accurate than w (without compromising on equity). Therefore, we would strictly prefer h over w, if we ignore interpretability requirements, and therefore we see that interpretability clashes with the desiderata of fairness and accuracy.

Remark. Note that unlike the previous result, the partition h does not guarantee an increase in equity. This makes sense, as we are now considering a more general notion of simplification.

Theorem 2. *For every non-trivial graded-simplification, say w, there exists a partition h (achievable by the features we have) that strictly improves accuracy and fairness (without hurting equity) with respect to w.*

Proof. Consider simplification w. It partitions X into the cells $C_1^\wedge, C_2^\wedge, \dots C_t^\wedge, \dots, C_d^\wedge$ with descending order of scores. Take a non trivial cell of w, say $C_t^\wedge$. Say $C_t^\wedge$ is the union of $C_a, C_b, \dots, C_z \in f'$.

Case 1: $V(C_t^{\wedge A}) \subseteq V(C_t^{\wedge D})$

There exists a cell C_a such that $C_a \in f'$, $C_a \subset C_t^\wedge$, such that C_a has the highest score amongst all cells $C_a, C_b, \dots, C_z \subset C_t^\wedge$ and only consists of disadvantaged instances.

Construct h as follows: Remove $\epsilon > 0$ mass of X from C_a to create a separate cell C'. Denote the remainder of $C_t^\wedge$ by C''. Observe that $S(C') > S(C_t^\wedge) > S(C'')$. Take ϵ small enough to not change order of C'' in the partition w (we can do this because of the genericity assumption). It should be in the same position as $C_t^\wedge$ was before. The only change in the order is that C' jumps to some position ahead of C''.

The new partition h is

$$C_1^\wedge, C_2^\wedge, \cdots, C_j^\wedge, C', C_{j+1}^\wedge, \cdots C_{t-1}^\wedge, C'', C_{t+1}^\wedge \cdots, C_d^\wedge$$

with descending order of scores.

Similar to the proof of Theorem 1, it is easy to check that for all rates r, the fairness and accuracy of h is at least as good as w, and for at least one value of r, strictly better in both criteria. We also see that the equity does not reduce.

Case 2: $V(C_t^{\wedge D}) \subseteq V(C_t^{\wedge A})$

There exists a cell C_a such that $C_a \in f'$, $C_a \subset C_t^\wedge$, such that C_a has the lowest score amongst all cells $C_a, C_b, \dots, C_z \subset C_t^\wedge$ and only consists of advantaged instances.

Construct h as follows: Remove $\epsilon > 0$ mass of X from C_a to create a separate cell C'. Denote the remainder of $C_t^\wedge$ by C''. Observe that $S(C') < S(C_t^\wedge) < S(C'')$. Take $\epsilon > 0$ small enough to not change order of C'' in the partition w. It should be in the same position as $C_t^\wedge$ was before (We can do this because of the genericity assumption). The only change in the order is that C' jumps to some position behind C''.

The new partition h is

$$C_1^\wedge, C_2^\wedge, \cdots, C_{t-1}^\wedge, C'', C_{t+1}^\wedge \cdots, C_v^\wedge, C', C_{v+1}^\wedge, \cdots C_d^\wedge$$

with descending order of scores.

Similar to the proof of Theorem 1, it is easy to check that for all rates r, the fairness and accuracy of h is at least as good as w, and for at least one value of r, strictly better in both criteria. We also see that the equity does not reduce.

□

Adding the Disadvantage Condition

In Theorem 2, if we make the disadvantage assumption, we can find a partition h that guarantees a strict increase in equity as well. That is, we get the following statement below. We omit the proof because essentially the same construction as in the main result of Kleinberg et al. [16] works for this result as well.

Theorem 3. *For every non-trivial graded-simplification, say w, there exists a partition h (achievable by the features we have) that strictly improves accuracy, fairness, and equity with respect to w (if we make the disadvantage assumption).*

Chapter 4

Trade-Offs between Fairness and Privacy

The result in this section essentially shows that there is no learning algorithm that is fair (even approximately), and differentially private, while maintaining good accuracy. Hence, we see that, the properties of fairness, differential privacy, and accuracy, are at odds with each other and it is not possible to satisfy the three of them simultaneously.

4.1 Setup

Notation To keep in line with previous work in this area, we use slightly different notation in this section as compared to the previous one.

Throughout, we use X to denote the domain set. There is probability distribution D over X. The domain set consists of elements of the form $z = (x, a, y)$, where x refers to the element's features (e.g., this could be income, name, etc.), a is a protected (binary) attribute (as before we have an advantaged and a disadvantaged group, and use $a = 0$ to denote the protected class). y is a binary label, that is the thing we want to predict. Additionally, throughout, we assume that $y = 0$ denotes the 'bad' label—meaning, for instance, in the context of, say, giving loans, this means that the person will not return the loan.

4.1.1 Differential Privacy

The notion of privacy we consider is called differential privacy. Differential privacy aims to protect the privacy of each individual in a database. In the case of learning algorithms, the database is the training set.

Database

We will first define what we mean by a database. We talk two different notions of a database.

The first one, is a finite sample, with entries drawn i.i.d. from the distribution D over the domain X. The second notion is to consider the whole distribution D as a database. The first notion is standard in the privacy literature, where databases are viewed as a finite collection of data points from n individuals. The second notion is standard for statistical notions of fairness, where the goal is to ensure fairness over a large population. Notion 2 can simply be considered a generalization of Notion 1. We will be using the second notion, but the same results and proofs also work for the first notion.

Neighbouring Databases

Given our definition of a database, it now remains to be defined what we mean by *neighboring databases*. Here we use the notion of σ-closeness as proposed by McGregor et al. [20], which is also used by Cummings et al. [2].

Definition 9 (σ-closeness [20])**.** Distributions D and D' are said be σ-close if

$$\frac{1}{2}\sum_{z\in X}|D(z) - D'(z)| \leq \sigma.$$

We calculate the distance between two distributions (databases) by the above expression (this is also known as total variation distance), and if the distance is lesser than σ, for some pre-specified value of σ, then the distributions are said to be neighboring.

Now that we have what it means for two databases to be neighboring, we can formally define differential privacy as shown below.

The Privacy Guarantee

Definition 10 ((ϵ, δ)-differential privacy [6])**.** For an $\epsilon, \delta \geq 0$, a randomized algorithm $\mathcal{A}$ is said to be (ϵ, δ)-differentially private if for all pairs of neighboring databases D, D' and for all sets $S \in \text{Range}(\mathcal{A})$ of outputs,

$$\Pr[\mathcal{A}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{A}(D') \in S] + \delta.$$

Differential privacy essentially ensures that an algorithm will generate similar outputs on neighboring databases (or distributions). It roughly protects the privacy of an individual in the database in the following way; changing an individual's entry, or deleting or adding it, will lead to what we call a neighboring database, and because the algorithm will generate similar outputs on neighboring databases, an observer seeing its output essentially cannot tell if a particular individual's information was used in the computation, or what that information is.

Remark. Although we have defined differential privacy in its full generality, note that throughout we will be talking about $(\epsilon, 0)$-differential privacy.

4.1.2 Fairness

What notion of fairness do we use? Essentially, our results hold for any reasonable notion of fairness, that do not allow one group to be treated much worse than the other. By "much worse," we mean, for example, high difference in true positive rates between the two groups, or some other sensible measure. Pretty much all the standard notions proposed fit this description. For example: Demographic Parity, Equal Opportunity, Equalised Odds, Calibration, etc. all work. These in turn are defined below. More importantly, even relaxed or approximate versions of these notions fit this description.

Definition 11 (Demographic parity)**.** A binary classifier h satisfies demographic parity if with respect to random variables A and Y

$$\Pr_{z \sim D}[h(z) = 1 | A = 1] = \Pr_{z \sim D}[h(z) = 1 | A = 0].$$

Definition 12 (Equal opportunity [13])**.** A binary classifier h satisfies equal opportunity if with respect to random variables A and Y

$$\Pr_{z \sim D}[h(z) = 1 | Y = 1, A = 1] = \Pr_{z \sim D}[h(z) = 1 | Y = 1, A = 0].$$

In words, h satisfies equal opportunity if it produces equal true positive rates across the two groups.

Definition 13 (Equalized odds [13])**.** A binary classifier h satisfies equalized odds if

- h has equal false positive rates across the two groups, i.e., with respect to random variables A and Y

$$\Pr_{z\sim D}[h(z) = 1|Y = 0, A = 1] = \Pr_{z\sim D}[h(z) = 1|Y = 0, A = 0]$$

- h satisfies equal opportunity.

4.2 Main Result

As stated previously, our main result is an incompatibility theorem showing how differential privacy and fairness are at odds with each other when we consider a learning algorithm with non-trivial accuracy. In particular, we consider the task of learning a classifier for a simple binary classification setting even when the learning algorithm is given full access to the underlying distribution and show that any learning algorithm that is $(\epsilon, 0)$-differentially private, and even approximately fair, cannot achieve accuracy better than that of a constant classifier. Note that the theorem also holds for the case where the algorithm has access to a finite training set, and not the whole distribution.

Before we state and prove our result, as mentioned previously in Section 2, our result here is a stronger version to one claimed in a paper by Cummings et al. [2], but as mentioned there, we believe that their proof has a gap. Below we first briefly describe what this gap is, and subsequently we move on to our theorem.

Gap in Proof in Previous Work [2, Theorem 1]. We noticed that we couldn't proceed with the argument as mentioned in the proof of Theorem 1 in Section 3. On a high level, what their proof tries to do is, given a distribution D and a classifier h that satisfies equal opportunity for this distribution, to essentially construct a distribution D' on which h does not satisfy equal opportunity. However, to the best of our understanding, there is error here (and in particular in the line claiming "...h does not satisfy equal opportunity with respect to D'") since h does satisfy equal opportunity on the D'. The error seems to stem from an incorrect usage of conditional probability arguments, and unfortunately this error does not seem fixable within the same proof idea. In any case, we do think that the statement is correct, and as we will show next, we can show a stronger claim.

Theorem 4. *If a learning algorithm $\mathcal{A}$ is $(\epsilon, 0)$-differentially private and is guaranteed to output an approximately fair classifier, then $\mathcal{A} : \mathcal{D} \to \Delta(\mathcal{H})$, where $\mathcal{D}$ denotes the set of all distributions, and*

$$\mathcal{H} = \{h : \mathcal{X} \to \{0, 1\} \mid h \text{ is a constant function}\}.$$

Before we present a formal proof, we start with an informal overview of the main idea. The main idea in the proof is to first observe that, due to differential privacy constraints, if there is a classifier that is output with positive probability by $\mathcal{A}$ on a distribution $D_1 \in \mathcal{D}$, then $\mathcal{A}$ has to output this classifier with positive probability on any other distribution $D_1' \in \mathcal{D}$. Now, what the claim above implies is that, if algorithm $\mathcal{A}$ has to be fair as well, and it outputs h on some input, then h is always fair, irrespective of the underlying distribution. Now, once we have the observation above, then it just remains to show that such classifiers—i.e., ones that are fair with respect to any underlying distribution—belong to a very restricted set, namely $\mathcal{H}$ as defined in the statement of theorem.

This concludes the overview. Below, we present a formal argument by first proving the following claim.

Claim 5. *Let $\mathcal{A}$ be a learning algorithm that is $(\epsilon, 0)$-differentially private. Then, $\forall D_1, D_1'$ $\in \mathcal{D}$, and for all classifiers h,*

$$\Pr[\mathcal{A}(D_1) = h] > 0 \implies \Pr[\mathcal{A}(D_1') = h] > 0.$$

Proof. Consider an arbitrary distribution $D_1 \in \mathcal{D}$ and an arbitrary classifier h such that $\Pr[A(D_1) = h] > 0$. Next, consider any arbitrary distribution $D_1' \in \mathcal{D}$. We need to show that $\Pr[\mathcal{A}(D_1') = h] > 0$.

To see this, first let us consider, for any $i \in [n]$ and $\eta > 0$, two η-close distributions D_i and D_{i+1} (i.e., they are neighboring databases). Since $\mathcal{A}$ is ϵ-differentially private, if $\Pr[\mathcal{A}(D_i) = h] > 0$, then we have that $\Pr[\mathcal{A}(D_{i+1}) = h] > 0$, for if otherwise, then we have,

$$0 < \Pr[A(D_i) = h] \leq \exp(\epsilon) \Pr[A(D_{i+1}) = h] = 0,$$

which is a contradiction.

Now, given the observation above, observe that, for any $\eta > 0$, one can construct a (finite) series of distributions $D_2, \cdots, D_n$ such that $\forall i \in [n]$, D_i and D_{i+1} are η-close (i.e.,

they are neighboring databases) and where $D_{n+1} = D'_1$. This in turn implies that we have,

$$\begin{aligned}\Pr[A(D_1) = h] > 0 &\implies \Pr[A(D_2) = h] > 0 \\ &\implies \Pr[A(D_2) = h] > 0 \\ &\vdots \\ &\implies \Pr[A(D_{n+1}) = h] > 0,\end{aligned}$$

where all the implications above are obtained by using the argument made above that for two neighboring databases D_i and D_{i+1}, $\Pr[\mathcal{A}(D_i) = h] > 0 \implies \Pr[\mathcal{A}(D_{i+1}) = h] > 0$. This in turn proves our claim. □

Equipped with the claim above, we are now ready to show the proof of our theorem.

Proof of Theorem 4. From Claim 5 we know that if a learning algorithm $\mathcal{A}$ is $(\epsilon, 0)$-differentially private and is guaranteed to output a fair classifier, then for all fair classifiers h and $\forall D_1, D'_1 \in \mathcal{D}$, $\Pr[\mathcal{A}(D_1) = h] > 0 \implies \Pr[\mathcal{A}(D'_1) = h] > 0$. In other words, what this implies is that, for a fair learning algorithm $\mathcal{A}$, any fair classifier h that is output by $\mathcal{A}$ is fair with respect to any distribution in $\mathcal{D}$. Below, we show how any h satisfying the property mentioned above should belong to $\mathcal{H}$, where $\mathcal{H}$ is as defined in the statement of the theorem.

To do this, consider for the sake of contradiction any $h \notin \mathcal{H}$. This implies that, for $y_1, y_2 \in \{0, 1\}$, there exist points $p_1 = (x_1, 0, y_1)$ and $p_2 = (x_2, 1, y_2)$ such that, either

1. $h(p_1) = 0$ and $h(p_2) = 1$, or
2. $h(p_1) = 1$ and $h(p_2) = 0$.

Now, if this is the case, then we will construct a distribution on which h is unfair. We construct a distribution for Case 1. To construct such a distribution, let us first consider the following points.

$$\begin{aligned}q_1 &= (x_1, 0, 1) \\ q_2 &= (x_2, 1, 0)\end{aligned}$$

Next, let us define the following distribution D'.

$$\begin{aligned}D'(q_1) &= \frac{1}{2} \\ D'(q_2) &= \frac{1}{2}\end{aligned}$$

Note that $h(q_1) = 0$ and $h(q_2) = 1$. However, if this is the case, then note that by any reasonable notion of fairness, h is unfair to group 0 as compared to group 1, since group 0 always has true label 1 but is always labeled 0, whereas group 1 always has true label 0 but is always labeled 1.

We omit the construction for Case 2, as essentially the same idea as Case 1 can be applied to Case 2 as well. □

4.3 Other Directions

As mentioned in Section 2, Kuppam et al. [18] consider scenarios in which personal data collected about individuals (e.g., census data) is used to decide the allocation of funds or resources. Because of privacy concerns, noise is added to the data in such a way that the queries on it satisfy ϵ-differential privacy. In this setting, Kuppam et al. [18] show, through empirical analysis, that this process of adding noise to the data often leads to 'unfairness', i.e. disproportionately impacts some groups over others.

One direction that we believe is promising, and which in fact was the direction we were initially planning to pursue, is to abstract the phenomena observed in Kuppam et al. [18] and come up with a framework that captures the situation more concretely. Within this framework, we hope to try and identify the sources of unfairness and back it with theoretical justification, and study more carefully the trade-offs between privacy and fairness. Note that although our result is a strong impossibility, it only says how one cannot always hope for fairness and differential privacy to hold together in the case of learning algorithms with non-trivial accuracy, and so this does not preclude the existence of reasonably fair and private algorithms for specific applications like that in Kuppam et al. [18].

We believe that this can be done by considering solutions that can broadly be classified into two categories based on their approach:

1. Analysing the step where we add noise to the data, and trying to modify this step so that it leads to less unfairness, while still giving similar privacy guarantees. This could potentially involve coming up with alternative differentially private algorithms—and especially in the context of our result, (ϵ, δ)-differentially private algorithms for $\delta > 0$—that are more constrained in the way they add noise, and is therefore in line with the existing research on differential privacy. This could also involve alternative privacy preserving techniques which do not satisfy differential privacy.

2. Analysing the step where we use the modified data to make decisions. That is, we could analyze the allocation algorithms that are used and modify them in such a way that they lead to less unfairness by taking into consideration the fact that the data is noisy (and hence uncertain). This in turn is more in line with the literature on fair division that has been extensively pursued for about a century in economics and more recently in computer science as computational social choice (e.g., [1, 22, 23]).

We elaborate our thoughts regarding both of these two approaches in the subsections below.

4.3.1 Understanding the Need for, and Looking at Alternatives to, Differential Privacy

As a first step towards analysing if we can have modify existing differentially private algorithms or suggest new alternatives, we need to understand [12] why differential privacy is a requirement in the settings like the ones described in Kuppam et al. [18], since, as also noted by Mervis et al. [21], it is possible that differential privacy may be too strong a technique for some scenarios.

To see this, consider the following naive method, which at least at first glance seems to satisfy the privacy needs described in Kuppam et al. [18] and also gives more accurate and more fair outcomes. Consider a database, which is represented as a table. Each row corresponds to a particular person, and each column corresponds to an attribute or feature (such as age or income) of that person. Now we anonymise the data (remove names etc. to protect against membership attack) and remove outliers for an attribute (let's say we remove Bill Gates from the table because his net worth is more than 100 Billion). Let us say there are m rows and n columns. Now, consider a specific column C_i and choose a random permutation π_i of $[1, 2, \cdots, m]$ for that column, and permute the entries of that column according to the chosen permutation. Similarly, for every column C_j choose a random permutation π_j of $[1, 2, \cdots, m]$, and permute the entries of C_j according to the chosen permutation. We now publish this modified table.

We can now get almost exactly accurate and fair answers to the queries we might have over single attributes (e.g., average income, number of people in a certain age group, etc.). However, we note that we cannot get accurate answers to queries over multiple attributes (e.g., average income of people in a certain age group, etc.). Although this clearly limits the utility of this method, we would like to argue that most of the scenarios described in Kuppam et al. [18] seem to consider only queries over single attributes. It would also

therefore be reasonable to assume that such single attribute queries would be useful in a real world setting, and a refinement of this naive model for these queries, or in general alternatives to differential privacy, is a possible thing to explore.

4.3.2 New Algorithms for Resource Allocation

Another direction we believe is promising is to look at the issue of resource allocation under input uncertainty. To be more concrete, consider the output of a differentially private algorithm as a random variable. Since we know its posterior distribution, we can think of the inputs to the resource allocation problems that are described in Kuppam et al. [18] as just being uncertain and try to come up with algorithms that take this uncertainty into account. This would likely involve showing that some algorithms can give better worst-case or average-case bounds (in terms of the regret) and we believe that this is related to some work, for example in mechanism design [10], that explores designing mechanisms under input uncertainty.

In summary, despite the result in this work, we do think that there is much scope in terms of the kind of questions or directions one can explore to better understand the trade-offs between privacy and fairness.

Chapter 5

Conclusion and Future Work

Through this work, we see that in decision making algorithms, the desiderata of fairness, interpretability, and privacy may be at odds with each other and it is often necessary to make trade-offs between them, if we want to maintain accuracy. We prove two theoretical results to demonstrate this.

The first main result considers a formal framework to build interpretable classifiers by 'simplicity', and shows that if we try to restrict our classifier to be simple within this framework, it can be replaced by a more complex classifier that strictly improves both fairness and accuracy. Therefore, we see that simplicity/interpretability clashes with the properties of fairness and accuracy.

There are many variants of the setup for the first main result that we could investigate for further work. While this result talks about the tradeoffs between fairness and simplicity, it is important to note that not all forms of simplicity (for e.g., linear classifiers) are captured by this framework. It would be interesting to investigate the compatibility between fairness and other notions of simplicity. Also, we deploy a particular objective function to quantify unfairness, and it might be worth looking into the interplay between interpretability and fairness for other fairness objectives.

The second main result is an incompatibility theorem showing how differential privacy and fairness are at odds with each other when we consider a learning algorithm with non-trivial accuracy. In particular, we consider the task of learning a classifier for a simple binary classification setting and show that any learning algorithm that is $(\epsilon, 0)$-differentially private, and even approximately fair, cannot achieve accuracy better than that of a constant classifier.

In the second result, the current statement allows the the learning algorithm to be faced with any underlying distribution (without any restrictions). But in reality, it's probably more likely that the set of distributions the learning algorithm will encounter follow some niceness properties. So, if we restrict the distributions by these niceness properties, can we prove something similar?

Another interesting direction of work could be to look at situations where one would want to have both interpretability, and privacy, and study the trade-offs these two requirements.

References

[1] Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[2] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Fairness in User Modeling, Adaptation and Personalization (FairUMAP 2019)*, 2019.

[3] Jennifer L Doleac and Benjamin Hansen. Does "ban the box" help or hurt low-skilled workers? statistical discrimination and employment outcomes when criminal histories are hidden. Technical report, National Bureau of Economic Research, 2016.

[4] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[5] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[7] Cynthia Dwork and Deirdre K Mulligan. It's not privacy, and it's not fair. *Stan. L. Rev. Online*, 66:35, 2013.

[8] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[9] Michael D Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. Privacy for all: Ensuring fair and equitable privacy protections. In *Conference on Fairness, Accountability and Transparency*, pages 35–47, 2018.

[10] Uriel Feige and Moshe Tennenholtz. Mechanism design with uncertain inputs:(to err is human, to forgive divine). In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 549–558. ACM, 2011.

[11] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[12] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *ACM Queue*, 2019.

[13] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[14] Marisa Hotchkiss and Jessica Phelan. Uses of census bureau data in federal funds distribution. *US Dept. of Commerce, Econ. and Statistics Administration*, 2017.

[15] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. *arXiv preprint arXiv:1812.02696*, 2018.

[16] Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: Implications for fairness, stereotypes, and interpretability. Technical report, National Bureau of Economic Research, 2019.

[17] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[18] Satya Kuppam, Ryan McKenna, David Pujol, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair decision making using privacy-protected data. *CoRR*, abs/1905.12744, 2019.

[19] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

[20] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE, 2010.

[21] Jeffrey Mervis. Can a set of equations keep us census data private. *Science Magazine*, 2019.

[22] Ariel D Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

[23] Jack Robertson and William Webb. *Cake-Cutting Algorithms: Be Fair If You Can*. A.K. Peters., 1998.

[24] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.