ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Algorithmic Fairness Revisited

Florian Tramèr

School of Computer and Communication Sciences

Master Thesis

August 14, 2015

Supervised by
Prof. J-P Hubaux
EPFL / LCA1

Abstract

We study *fairness in algorithmic decision making*, with the goal of proposing formal and robust definitions and measures of an algorithm's bias towards sensitive user features. The main contribution of this work is a statistical framework for reasoning about equity in algorithmic decisions, while also considering various constraints on users' and the algorithm vendors' utilities. We first revisit previous notions of fairness from the literature, that are based on different measures of the dependency between sensitive features and algorithmic decisions. We illustrate several limitations of these measures, such as their failure to generalize to non-binary sensitive features or algorithm outputs, and we propose a more general and robust fairness measure based on *mutual information*, which has received little attention so far. In particular, we show that our fairness measure produces significantly better characterizations of the *statistical significance* of an algorithm's bias, compared to the notion of *statistical parity* introduced by Dwork et al. [17, 73].

We further discuss the inadequacy of previously considered fairness measures, in their inability to detect large-scale discriminatory practices, that are due to algorithms with small biases being applied on a global scale. We instigate the discussion on *statistical hypothesis tests*, that, in spite of being standard tools in legal practices, have received little attention in the context of algorithmic fairness. In this regard, we present another advantage of mutual information, compared to other proposed fairness measures, in that it is directly linked to a popular statistical goodness-of-fit test known as the *G-test*.

We further reason about situations, where the absolute parity of an algorithm may be prohibitively at odds with the *utility* of an algorithm's vendor or its users. We generalize our fairness definitions to include various utilitarian constraints, with a particular focus on discriminatory practices that are considered acceptable because of genuine *business necessity requirements*. We describe a framework mirroring legal practices, that allows businesses to discriminate users based on genuine *task-specific qualification levels*, in order to guarantee the organization's well-being.

Finally, we consider practical issues related to the detection of algorithmic biases from empirical data, and propose a generic methodology, relying on cluster analysis techniques and robust statistical testing, to reason about discrimination in different subsets of the user population. We evaluate our methods on small artificial datasets, as well as on the *Berkeley Graduate Admissions* and *Adult Census* datasets, and illustrate how our techniques can either discover discrimination hidden in particular user subsets, or reveal potential business-necessary requirements that may account for an observed algorithmic bias.

Contents

1 Introduction

With the rapid growth of information systems that collect and mine user data, an increasing portion of people's lives is becoming dictated by algorithmic decision making. While this accumulation of digital personalized data may benefit both companies and individuals, through improved consumer models and personalized treatments for instance, it also harbors the potential for discrimination on an unprecedentedly large scale. Indeed, following the ongoing erosion of digital privacy, where sensitive information about users is disclosed either directly or indirectly from their data, we are faced with the challenge of ensuring that these algorithmic decisions remain fair, and avoid exhibiting discriminatory biases towards features such as gender, race or age for instance [31].

It has long been recognized that simply removing an attribute regarded as sensitive from a user's data is insufficient, as a variety of other non-sensitive attributes may be correlated with it. For instance, the first name or ZIP code of a user may not be considered sensitive in a particular situation, yet they could be strongly linked to the user's ethnicity. An illustrative example of this phenomenon can be found in an investigation from the Wall Street Journal, that discovered that the online pricing scheme of Staples Inc, a famous office-supply chain store, varied depending on a user's location [71]. While a user's location may be regarded as a non-sensitive feature, and the pricing scheme appears unlikely to have been designed with a discriminatory purpose in mind, the journalists noticed that Staples' prices were typically higher in neighborhoods with low average income. The price setting algorithm was thus discovered to exhibit a discriminatory bias against a sensitive feature (income), that wasn't even part of the data that Staples collects about its users. A major challenge thus lies in designing mechanisms, that reason about correlations between sensitive features and algorithmic decisions, regardless of the ways in which the algorithm makes use of the sensitive data.

Additional difficulties arise from the fact that sensitive attributes may be linked to the utility that users or businesses derive from a particular algorithmic decision. Law and policymakers have long studied the intrinsic and sometimes conflicting relations between fairness and utility, in order to identify situations where one notion might be given priority over the other. A robust model for reasoning about algorithmic fairness will therefore also incorporate different notions of utility, both for users and businesses, and specify in which ways these concepts interact.

Fairness Measures In recent years, researchers from the data mining, privacy and machine learning communities have begun to investigate the problems of discovering [60, 58, 44, 74, 22] and preventing [36, 37, 12, 44, 74, 38, 17, 73, 30, 22] algorithmic discrimination. These lines of work have led to the introduction of various definitions of fairness, inspired by policy and legal practices. All these works define fairness in terms of the dependency, or relationship between an algorithm's outputs and a user's sensitive attributes, that can be estimated from empirical data. Previously considered measures can be divided into 3 main categories.

- *Ratio measures* that consider the ratio between the proportions of outcomes for two groups of users [60, 58, 17, 30, 22].
- *Difference measures* that consider the absolute difference between the proportions of outcomes for two groups of users [36, 37, 12, 60, 58, 44, 74, 17, 73].

- *Information theoretic measures* that consider the mutual information between the distributions of the outcomes and of the sensitive attributes [38].

After we formally introduce these measures, we will discuss possible extensions of ratio and difference measures from binary to multivalued sensitive features and algorithmic outputs, leading to issues that were overlooked by previous works. In regards to these limitations, we argue that mutual information provides a more general and robust measure of algorithmic bias, than other measures appearing in the literature.

As the problem we consider is that of quantifying and estimating an algorithm's fairness from empirical data, the *statistical significance* of our measurements also plays an important role. We identify a number of scenarios, where one or the other of the mentioned measures fails to detect or properly quantify discriminatory practices, if the statistical significance of the measurements is omitted. In particular, we show that *statistical parity* [17, 73] is not an appropriate measure of statistical dependency between random variables, and thus that it is ill-suited for reasoning about fairness detection and prevention in algorithmic decision making.

Previously considered fairness definitions such as a-protection [60, 58], ϵ-fairness [22] or statistical-parity [17] rank an algorithm as unfair, if some ratio- or difference-measure of the algorithm's bias exceeds a *policy-defined threshold.* Although such *threshold-fairness* definitions are naturally justified by current legal practices, where one or another method may be used to unveil discrimination on a case-by-case basis, we argue that they are ill-suited for reasoning about algorithmically-induced discrimination on a big-data scale. Indeed, we show that because they disregard the *absolute magnitude* of a discriminatory practice, such techniques fail to detect small inherent biases of algorithms, that are deployed and applied on a very large scale.

Another technique introduced in the legal field for reasoning about discrimination from empirical data is *statistical hypothesis testing.* We show that statistical tests of independence provide a formal methodology, for detecting and quantifying any kind of discriminatory bias in algorithmic decision making, by effectively measuring the likelihood, that algorithmic outputs are truly independent from sensitive features. In this setting, we identify an additional advantage of mutual information over other proposed measures of fairness, in that it immediately yields a robust statistical test known as the *G-test.* Using this connection, we present a formal definition and measure of algorithmic fairness, based upon an information-theoretical statistical test of the independence between sensitive features and algorithmic decisions.

Fairness and Utility A limiting aspect of fairness measures, that are solely based upon the dependency between sensitive features and algorithmic outputs, is that they are not applicable to situations, where correlations between an algorithm's decisions and a user's sensitive attributes may be either desirable or acceptable. We thus further investigate ways to extend our fairness definition to cases where either users' or business's utilities are at odds with absolute parity.

First of all, we consider scenarios where users may derive different utilities from an algorithm's outcomes. In ad-targeting for instance, it is reasonable to assume that a user's utility function will be highly correlated with sensitive attributes such as gender or age. Many previous definitions of algorithmic fairness [12, 60, 58, 36, 37, 74, 22] rely on the simplifying assumption that an algorithm's outputs are either inherently 'positive' or 'negative', and are therefore ill-suited for such cases. In contrast, we show how to generalize our fairness definitions to take into account user-utility in a straightforward way, by reasoning about the statistical dependency between sensitive attributes and the *utility* derived from an algorithm's decision, rather than the decision itself.

A different and more subtle issue has to do with situations, where absolute fairness may be prohibitively at odds with the utility of the business involved. Indeed, it might be the case that a user's sensitive attribute is highly correlated with the user's *qualification* for some job, or his *credit-score* for a new loan. There has been a vast amount of work in the legal community focusing on defining and understanding the notion of *business-necessity*, which enables justifying discriminatory practices in the presence of genuine utility requirements. We propose a flexible framework for defining and measuring fairness in the presence of business-necessity constraints, by considering *alternative null-hypotheses* for our statistical tests. Our approach mimics legal practice, and additionally increases transparency, in that it shifts the task of providing and justifying a business-utility requirement to the business itself. More precisely, we expect the business to specify a *utility function* that classifies users into *qualification levels*, based on a number of non-sensitive attributes. The question of whether this classification represents a genuine business requirement will depend on policy governing the algorithmic decision being considered. Given such a business-utility function, we show how to extend our fairness measures to detect inherent discriminatory practices that are not explained by the business's utility requirements.

Statistically Robust Discrimination Discovery Using the new fairness definitions and statistical testing framework that we propose, we further explore practical issues that must be overcome, in order to design a robust system for the detection and analysis of algorithmic biases from empirical data. We discuss different statistical fallacies, such as *Simpson's paradox* and the *multiple comparisons problem*, that might arise in the context of discrimination discovery, and we introduce a simple methodology, relying on cluster analysis techniques, to detect and reason about discrimination arising in different subsets of the users. We show how to identify easily interpretable user sub-populations, that can be defined in terms of only a small number of non-sensitive attributes, and that are treated differently with respect to the considered algorithmic task. Through a series of examples, we illustrate how these clusters may enable the detection of both discriminatory practices arising only in particular user sub-populations, as well as potential business requirements that account for an observed bias. Combined with robust statistical testing tools, our techniques yield results about the discriminatory biases of an algorithm, both in the population as a whole as well as in chosen sub-clusters, that can then be evaluated according to contextual policies.

We implement a prototype of our proposed mechanism, and evaluate it on different datasets including the famous 1973 Berkeley Graduate Admissions Data [9], as well as the Adult Census Dataset [1]. Our data-clustering and statistical testing framework produce results, that are consistent with previous analyses of these datasets from the literature, both from a statistical and sociological perspective.

[1] `https://archive.ics.uci.edu/ml/datasets/Adult`

Outline The rest of this work is organized as follows. In Section 2, we discuss algorithmic fairness in the absence of user or business utility constraints. We begin by introducing previously considered definitions of fairness in Sections 2.1 and 2.2, and illustrate various limitations of these approaches. We further focus on a fairness measure based on mutual information, and illustrate its various advantages over other measures appearing in the literature. We then propose a more robust statistical testing framework for discrimination discovery on a big data scale in Section 2.3. In Section 3, we extend our discussion to utility constraints, first for users (Section 3.1) and then for the business itself (Section 3.2). We review related prior work in Section 4. In Section 5, we discuss practical considerations of our methodology from a system perspective. In particular, we focus on issues related to data collection (Section 5.1), to the discovery of discrimination in data subsets (Section 5.2), and to the design of robust statistical tests (Section 5.3). Section 6 presents some experimental results, both on illustrative toy-examples, as well as on the Berkeley Admissions and Adult Census datasets. We conclude and expose open questions and directions for future work in Section 7. We provide some legal background on discrimination detection in Appendix A.

2 Non-Utilitarian Fairness

Previous notions of algorithmic fairness defined in the literature all agree on the intuitive idea that absolute fairness with respect to some sensitive attribute (in the absence of utility constraints), is obtained whenever the algorithm's output is *independent* of this attribute. The different measures of fairness differ in the way that the level of dependency between sensitive attributes and algorithmic decisions is quantified.

By solely considering interactions between some sensitive attribute and the algorithm's output, previous notions of fairness all view the algorithm as a *black-box*. As already mentioned, abstracting the inner workings of an algorithm is desirable, because the way in which an algorithm uses a sensitive attribute internally (if at all), does not necessarily reflect on the exhibited bias, as non-sensitive attributes correlated to sensitive ones can be the cause of discrimination.

2.1 Preliminaries and Notation

Throughout this work, a capital letter Y denotes a discrete random variable over some alphabet $\mathcal{Y}$. The distribution of Y is denoted $P(Y)$ and we write $P(y)$ as a shorthand for $P(Y = y)$, where $y \in \mathcal{Y}$. We denote statistical independence between Y and Z as $Y \perp Z$.

We consider users, whose collected data is represented by a (multidimensional) random variable $X \in \mathcal{X}$. Similarly, we denote a user's sensitive attribute(s) by a variable $S \in \mathcal{S}$. The data X may be correlated with the sensitive features S, as captured by a joint distribution $P(S, X)$, either because sensitive attributes are part of the collected data or because of inherent dependencies between certain sensitive and non-sensitive features. The decision-making algorithm $\mathcal{A}$ generates outputs $O \in \mathcal{O}$ by passing a user's data X through a conditional distribution $P(O \mid X)$.

Note that the true distribution $P(O \mid X)$ is unknown in general, since we assume black-box only access to $\mathcal{A}$. Thus, we will consider that we have access to some *user dataset* $\mathcal{D} = \{(x_1, s_1), (x_2, s_2), \dots, (x_N, s_N)\}$, of i.i.d. samples from $P(S, X)$. The correlations between S and O must then be estimated from a number of collected samples of the form (s_i, o_i), where o_i is the output produced by $\mathcal{A}$ on input x_i. We denote the empirical joint distribution of S and O over $\mathcal{D}$ by $\hat{P}(S, O)$. The marginal empirical distributions are then given by $\hat{P}(s) = \sum_{o \in \mathcal{O}} \hat{P}(s, o)$ and $\hat{P}(o) = \sum_{s \in \mathcal{S}} \hat{P}(s, o)$. Furthermore, the empirical conditional probability distribution of O given S is defined as $\hat{P}(o \mid s) = \frac{\hat{P}(s,o)}{\hat{P}(s)}$.

The general approach taken by previous definitions, inspired by legal practices, consists in selecting some measure of the dependence between the random variables S and O, and classifying the algorithm as fair or non-discriminatory if the measured *empirical* dependence between S and O over $\mathcal{D}$ falls below some policy-defined threshold α. We will refer to definitions of this form as 'threshold' fairness.

An issue with this approach, that we will illustrate through a series of examples, is that an algorithm, whose true bias falls below the policy-defined threshold, will be considered fair, even if it leads to significant discrepancies when applied on an very large scale. As algorithmic decision making is expected to become deployed and utilized in a global fashion, we believe that a more robust measure of fairness would consist in classifying an algorithm as unfair, whenever we can assess with high certainty that the algorithm exhibits a bias (however small it may be). Formally, in the absence of utility constraints, fairness is attained exactly when the algorithm produces outputs O truly independent of sensitive features S.

Definition 1 (Non-Utilitarian Fairness). *An algorithm $\mathcal{A}$ is fair with respect to a sensitive feature S, if and only if $O \perp S$.*

Given this definition, *measuring* the fairness of a given algorithm can be seen as an instance of a *statistical test of independence*, where our goal is to assess, from the empirical data, whether $\mathcal{A}$ exhibits any statistically significant bias or not. We will show that a fairness measure based on mutual information, that was previously proposed in [38], provides a nice link between ours and previously considered definitions of algorithmic fairness.

First of all, we will revisit the definitions of fairness from previous works, and illustrate multiple advantages of mutual information over difference- and ratio- based measures of an algorithm's bias. Secondly, we show that the mutual information measure can also be directly used in our hypothesis testing setting, through its natural connection to the G-test, a popular statistical goodness of fit test. In Section 3, we will consider more general scenarios, where independence between sensitive attributes and algorithm outputs may not be an acceptable characterization of fairness, because of user and business utility constraints.

2.2 Threshold-Fairness

We begin by introducing different measures of fairness that previously appeared in the literature. We first focus on the case of a binary sensitive attribute $S \in \{s^+, s^-\}$, as is the case in most of the prior work. It is usually assumed that one of the two attributes, s^+, represents the class of users 'favored' by the algorithm. We discuss possible issues when generalizing some fairness measures to multivalued S further on. For ratio- and difference-based measures, we use the terminology introduced by Ruggieri et al. [60, 58] based on the 'lift' measure, with some changes in notation. We focus on their $slift$ and $slift_d$ measures, which also appear in slightly different forms in [17, 22] and [36, 37, 12, 74, 17, 73]. The definitions from [60] also contain a *context of discrimination*, which we discuss in more detail in section 5.2.

Definition 2 (Ratio Measures). *For a sensitive attribute $S \in \{s^+, s^-\}$ and some output $o \in \mathcal{O}$, the selection-lift is defined as*

$$slift(s^+; o) = \frac{\hat{P}(o \mid s^+)}{\hat{P}(o \mid s^-)} \ .$$

Definition 3 (Difference Measures). *For a sensitive attribute $S \in \{s^+, s^-\}$ and some output $o \in \mathcal{O}$, the difference-based selection-lift is defined as*

$$slift_d(s^+; o) = \hat{P}(o \mid s^+) - \hat{P}(o \mid s^-) \ .$$

The ratio and difference selection-lifts capture the empirical ratio or difference, between the probability of seeing some output for users with or without a specific sensitive feature. To quantify the fairness of an algorithm $\mathcal{A}$, Ruggieri et al. [60, 58] measure the dependency between a sensitive attribute $S \in \{s^+, s^-\}$ and the proportion of outputs o, where o is a 'positive' output providing some benefit to users, and users with attribute s^+ are favored.

Definition 4 (a-protection)**.** *Let $f()$ be either the $slift$ or $slift_d$ measure, and $a \in \mathbb{R}$ a fixed threshold. Then $\mathcal{A}$ is a-protective w.r.t $f()$, $s^+ \in \mathcal{S}$ and $o \in \mathcal{O}$, if $f(s^+, o) < a$. Otherwise $\mathcal{A}$ is a-discriminatory.*

In the US, the so-called *four-fifths rule*, appearing in the Uniform Guidelines on Employee Selection Procedures (UGESP) issued by the Equal Employment Opportunity Commission (EEOC), can be seen as a specific instance of the a-protection definition. The rule states that a ratio of selection-rates (or $slift$) greater than $a = 1.25$ will 'generally be regarded [...] as evidence of adverse impact' [13]. Note that the $slift$ and $slift_d$ measures may yield very different conclusions on whether a mechanism is discriminatory or not. For instance, for small enough conditional probabilities, we may have $slift_d \approx 0$, but $slift$ arbitrarily high.

We now consider the mutual information measure used in [38]. In order to obtain a coherent set of definitions, we first introduce a measure based on the Kullback-Leibler divergence between $P(S)$ and $P(S \mid O = o)$ for a particular output $o \in \mathcal{O}$. The mutual-information $I(S; O)$ between the sensitive features and algorithm outputs is then defined as the expected value of this KL divergence over all outputs $o \in \mathcal{O}$.

Definition 5 (Information Theoretic Measures)**.** *For a sensitive attribute $S \in \mathcal{S}$ and some output $o \in \mathcal{O}$, the Kullback-Leibler divergence between $\hat{P}(S)$ and $\hat{P}(S \mid O = o)$ is defined as*

$$D_{KL}(\hat{P}(S \mid O = o) \mid\mid \hat{P}(S)) = \sum_s \hat{P}(s \mid o) \ln \frac{\hat{P}(s \mid o)}{\hat{P}(s)} \ .$$

While $slift$ and $slift_d$ intuitively capture how close $\hat{P}(O = o \mid S)$ is to $\hat{P}(O = o)$, the KL divergence instead measures the distance between $\hat{P}(S \mid O = o)$ and $\hat{P}(S)$. The two approaches are directly connected through Bayes' theorem, and both are valid characterizations of the empirical dependency relation between S and O as shown in the following theorem.

Theorem 6. *For a sensitive attribute $S \in \{s^+, s^-\}$, the following implications hold.*

$$\begin{aligned} \hat{P}(S, O) = \hat{P}(S)\hat{P}(O) &\iff slift(s^+; o) = 1, \ \forall o \in \mathcal{O} \\ &\iff slift_d(s^+; o) = 0, \ \forall o \in \mathcal{O} \\ &\iff D_{KL}(\hat{P}(S \mid O = o) \mid\mid \hat{P}(S)) = 0, \ \forall o \in \mathcal{O} \ . \end{aligned}$$

Thus, all the measures we have introduced somehow characterize the *empirical independence* between S and O. While Theorem 6 shows that these methods are equivalent in the limit where O and S appear independent, we are also interested in assessing the (in)fairness of a particular algorithm $\mathcal{A}$ when complete independence is not attained on the sampled data. We now discuss and illustrate various issues arising from Definitions 2-5 when we attempt to extend them to multivalued sensitive features, or when estimating and quantifying the dependency that $\mathcal{A}$ introduces between S and O.

2.2.1 Limitations of Previous Definitions

Symmetry The ratio- and difference-based measures introduced in [60] are inherently non-symmetric, in the sense that $slift(s^+; o) \neq slift(s^-; o)$ in general (and similarly for $slift_d$).

Most previous works [12, 60, 58, 36, 37, 74, 22] assume that the fairness measure is with respect to some *minority* or *protected-by-law group* with sensitive attribute $S = s^-$, and that the considered output $o \in \mathcal{O}$ is 'positive'.

An issue with this approach is that it might not be known *a priori* whether $\mathcal{A}$ might discriminate against one group of users or the other. For a binary feature such as gender, for instance, we would like to detect both discrimination against women or men, with equal quantification in both cases. The notion of statistical parity introduced in [17, 73] considers the absolute value of $slift_d$ to avoid this problem. An equivalent solution for the ratio measure would be to consider the maximal value between $slift(s^+; o)$ and $slift(s^-; o)$ as in [17]. Note that the Kullback-Leibler divergence metric is symmetric by definition.

Moreover, classifying outputs $o \in \mathcal{O}$ as either 'positive' or 'negative' might not be straightforward in general either. We discuss the problem of measuring the fairness of our algorithm over multiple output values further on. Furthermore, in Section 3.1, we consider situations where the *utility* perceived for a particular outcome may depend on sensitive features.

Multivalued Sensitive Features Ruggieri et al. [60] propose to extend their measures to multivalued sensitive attributes $S \in \{v_0, v_1, \ldots, v_d\}$, by comparing $\hat{P}(o \mid v_i)$ to $\max_k \hat{P}(o \mid v_k)$, where o is a positive output. Thus, we compare users with sensitive feature v_i to the users with the *most-favorable* feature for obtaining output o. Again, this approach assumes that we may classify outputs as either positive or negative a priori. With this approach, we obtain d different measures of the algorithm's bias, for each possible value taken by S.

Other works in which the $slift$ or $slift_d$ measures appear, have limited their analysis to the particular case of a binary sensitive feature [36, 37, 12, 60, 58, 74, 17, 73, 22]. Kamiran et al. [36] suggest that a multivalued sensitive attribute $S \in \{v_0, v_1, \ldots, v_d\}$ can always be transformed into a binary attribute $S' \in \{v_0, \bar{v_0}\}$, where $S' = \bar{v_0}$ when $S \neq v_0$. Such a transformation may however fail to uncover discriminatory practices, as shown in the following example. Let $S = \{$*black*, *white*, *hispanic*$\}$ and assume we are concerned about discrimination against black users. Suppose we use the transformation $S' = \{$*black*, *not-black*$\}$, and consider the data from Table 1, obtained from an algorithm deciding if a user should be hired or not.

S					
Black		White		Hispanic	
Applicants	Hired	Applicants	Hired	Applicants	Hired
100	50%	100	**80%**	100	20%

S'			
Black		Not Black	
Applicants	Hired	Applicants	Hired
100	**50%**	200	**50%**

Table 1: Transformation from multivalued to binary sensitive features, that hides discriminatory practices.

As we can see, after transformation, the output O appears completely independent of S'.

However, the algorithm is not fair with respect to race since it discriminates against both black and Hispanic users. To avoid this problem, we could obviously consider *all* possible binary transformations of the sensitive features. However, we would ideally like to obtain a single measure of an algorithm's fairness, rather than one for each possible value of the sensitive feature. In this sense, note that in contrast to the $slift$ and $slift_d$ measures, the KL divergence measure directly allows for multivalued sensitive features S.

Extensions to Multivalued Outputs As we are interested in measuring the amount of dependency that $\mathcal{A}$ introduces between S and O, we should consider some aggregation of the measures from Definitions 2-5 for all possible outputs $o \in \mathcal{O}$. Many previous works [12, 60, 58, 36, 37, 74, 22] consider a simplified setting with binary output $o \in \{o^+, o^-\}$, where one output is assumed to be inherently positive. The fairness of an algorithm is then simply assessed through either of the $slift(s^+, o^+)$ and $slift_d(s^+, o^+)$ measures.

Dwork et al. [17] and Zemel et al. [73] introduce a more general notion called statistical-parity, that aggregates $slift_d$ measures by summing over all possible outputs.

Definition 7 (Statistical Parity). *For a sensitive attribute $S \in \{s^+, s^-\}$, an algorithm $\mathcal{A}$ empirically satisfies statistical parity up to bias ϵ if*

$$D_{TV}(\hat{P}(O \mid S = s^+), \hat{P}(O \mid S = s^-)) = \frac{1}{2} \sum_{o \in \mathcal{O}} \left| \hat{P}(o \mid s^+) - \hat{P}(o \mid s^-) \right| \leq \epsilon \,,$$

where $D_{TV}(P, Q)$ is the total-variation distance between distributions P and Q.

This definition is limited to the case of a binary sensitive feature. Note that when O is itself a binary variable, statistical parity reduces to the absolute value of the $slift_d$ measure. We will see in Section 2.2.2, that when O is multivalued, the total-variation distance does not seem to be an adequate measure of the statistical significance of the dependency between O and S, and thus should be avoided as a measure of algorithmic fairness.

In addition to being easily extendable to multivalued sensitive features, the Kullback-Leibler divergence can also be generalized to multiple outputs, yielding the mutual information measure between O and S, appearing for instance in [38].

Definition 8 (Mutual Information). *For an algorithm $\mathcal{A}$, the empirical mutual information between S and O is defined as*

$$\hat{I}(S; O) = \mathbb{E}_O[D_{KL}(\hat{P}(S \mid O = o) \mid\mid \hat{P}(S))] = \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \hat{P}(s, o) \ln \left(\frac{\hat{P}(s, o)}{\hat{P}(s)\hat{P}(o)} \right) .$$

Intuitively, mutual information is a measure of two variables' mutual dependence. It quantifies how much information the sensitive feature S provides about the algorithm's output O. Mutual information can be normalized (see [19]) as $\hat{I}_{norm}(S; O) = \frac{\hat{I}(S;O)}{\min\left\{\hat{H}(S), \hat{H}(O)\right\}} \in [0, 1]$, where $\hat{H}(Y)$ is the empirical entropy of a random variable Y. In this form, $\hat{I}_{norm}(S; O)$ represents the proportion of information that one variable provides about the other, relative to the minimum entropy of the two. An interesting property of mutual information is its symmetry, $\hat{I}(S; O) = \hat{I}(O; S)$. This implies that we may reason about an algorithm's fairness both in terms of what the sensitive feature tells about the outputs, as well as what the outputs reveal about sensitive features, a problem seemingly related to privacy protection. This apparent *duality* between fairness and privacy has been noted by a number of researchers, and we discuss its implications in more detail in Section 4.

2.2.2 MI-Fairness

In a similar fashion to a-protection or statistical parity, we can define a 'thresholded' notion of fairness from mutual information.

Definition 9 (MI-Fairness)**.** *With respect to a sensitive attribute S, an algorithm $\mathcal{A}$ empirically satisfies ϵ-MI fairness if $\hat{I}(S;O) \leq \epsilon$, and ϵ-normalized MI fairness if $\hat{I}_{norm}(S;O) \leq \epsilon$.*

As we have seen, a main advantage of MI-fairness compared to other definitions of threshold-fairness is that it immediately handles multivalued sensitive features and algorithm outputs. Furthermore, no prior assumptions on the identity of the discriminated group or on the relative utilities of the algorithm's output is required. Finally, we now also show that MI-fairness produces more consistent characterizations of the empirical dependency between S and O than the statistical parity measure from [17, 73].

Comparing Statistical Parity and MI-Fairness We present a simple example illustrating the limitations of the total-variation measure, in appropriately characterizing the level of empirical dependency between S and O. Consider algorithms $\mathcal{A}$, $\mathcal{A}'$ that decide which users to promote in a company. Both algorithms output a value $o \in \{\textit{President}, \textit{Manager}, \textit{Employee}\}$ to indicate whether the user will be promoted to president, to manager or remain a normal employee. The sensitive attribute we consider is gender. Table 2 displays two sampled data-sets of algorithmic decisions by $\mathcal{A}$ or $\mathcal{A}'$ over 40,000 employees.

$\mathcal{A}$	Male	Female
President	20	5
Manager	9,980	9,995
Employee	10,000	10,000

$$D_{TV} = 7.50 \cdot 10^{-4}$$
$$\hat{I}_{norm} = 1.74 \cdot 10^{-4}$$

$\mathcal{A}'$	Male	Female
President	20	20
Manager	9,970	9,995
Employee	10,010	9,985

$$D_{TV} = 1.25 \cdot 10^{-3}$$
$$\hat{I}_{norm} = 1.13 \cdot 10^{-6}$$

Table 2: Comparison between total variation and mutual information as measures of fairness. D_{TV} denotes the total variation distance as in Definition 7 and $\hat{I}_{norm}$ denotes the normalized mutual information as in Definition 9.

The total variation measure yields a lower value for $\mathcal{A}$ than for $\mathcal{A}'$, yet $\mathcal{A}'$ intuitively seems much more fair than $\mathcal{A}$. Indeed, the small differences in proportions introduced by $\mathcal{A}'$ are much more likely to have been introduced by randomness or noise due to sampling, compared to the ones from $\mathcal{A}$. We see here that simply summing up $slift_d$ measures fails to take into account how significant a difference in proportion is, and thus does not produce an an appropriate measure of the statistical significance of the empirical dependency between the variables S and O, at least for our purpose.

We will see in Section 2.3, when we introduce the statistical G-test, that for datasets of fixed size, mutual information is directly linked to (an approximation of) the likelihood that the considered algorithm is unbiased. In the above example, $\mathcal{A}'$ is likely to be unbiased, with small differences explained by the randomness of the sample, while $\mathcal{A}$ appears to be biased with high certainty, given the observed data [2].

Therefore, when using the total variation as a measure of fairness ([17]) or as a quantity to optimize when training a fair classifier ([73]), there is a risk of considering algorithms with a clear bias such as $\mathcal{A}$ as more fair than algorithms such as $\mathcal{A}'$, that intuitively appear unbiased. Finally, we note that Dwork et al. [17] also introduce the D_∞ measure, which can be seen as an analog to D_{TV} where $slift_d$ is replaced by $slift$. We can easily construct similar examples to the one above, for which D_∞ would yield results incompatible with mutual information.

2.2.3 Statistical Significance and Hypothesis Testing

Having discussed different 'threshold' measures of fairness, we now illustrate their limitations when considering the notion of fairness given in Definition 1. As noted by Peresie, a concern with setting thresholds for bias measures is that this introduces a 'permissible level of discrimination' [54], meaning that a process might consistently exhibit a bias slightly below the threshold, and still be considered as fair.

As a motivating example, consider an algorithm $\mathcal{A}$ that decides whether a user will be hired or not. The sensitive attribute is gender, with half of the users expected to be male and half female. The algorithm is inherently biased, in that it systematically hires 51% of the male candidates it sees and only 49% of the female candidates (any other small bias could be considered). Note that while $\mathcal{A}$ exhibits only a small bias, it is still *discriminatory*, in the sense of Definition 1, since its outputs explicitly depend on sensitive features. Unless this bias can be explained by user- or business-utility considerations as discussed in Section 3, the algorithm should still be classified as *unfair*, if its inherent bias leads to large imparities when applied on a global scale.

To illustrate this, we sample random datasets $\mathcal{D}$ of size 200, 20,000 and 2,000,000 and report the values of the $slift(\texttt{male}, \texttt{hired})$, $slift_d(\texttt{male}, \texttt{hired})$ and $\hat{I}(S; O)$ measures in Table 3. The true measures of I, $slift$ and $slift_d$ are as follows.

$$I = 2.0 \cdot 10^{-4}, \quad slift = 1.041, \quad slift_d = 0.02\ .$$

When empirical data is used to measure the discrimination of an algorithm, Ruggieri et al. [60] suggest adding *confidence intervals* to the $slift$ and $slift_d$ measures, in order to get a better indication of the statistical significance of these values. They provide a more robust definition of a-protection, again for $S \in \{s^+, s^-\}$ and some positive output o for which users with attribute s^+ are favored.

Definition 10 (a-protection)**.** *Let $f()$ be either the $slift$ or $slift_d$ measure, and $a \in \mathbb{R}$ a fixed threshold. Denote the confidence interval of $f(s^+, o)$ at significance level β as $[L_1, L_2]$. Then $\mathcal{A}$ is a-protective at significance level β w.r.t $f()$, s^+ and o, if $L_2 < a$. $\mathcal{A}$ is a-discriminatory at significance level β if $L_1 \geq a$.*

[2] Specifically, the G-test yields a *p-value* (the probability that a fair algorithm would produce results as extreme as the ones observed), of 0.97 for $\mathcal{A}'$, and of 0.008 for $\mathcal{A}$

	Male		Female	
$\lvert\mathcal{D}\rvert$	Applicants	Hired	Applicants	Hired
200	100	**54.00%**	100	51.00%
20,000	10,015	**51.01%**	9,985	48.87%
2,000,000	1,000,941	**51.07%**	999,059	49.00%

$\lvert\mathcal{D}\rvert$	$\hat{I}$	$slift$	$slift_d$
200	$4.512 \cdot 10^{-4}$	1.059 ± 0.264	0.030 ± 0.138
20,000	$2.290 \cdot 10^{-4}$	1.044 ± 0.028	0.021 ± 0.014
2,000,000	$2.143 \cdot 10^{-4}$	1.042 ± 0.003	0.021 ± 0.001

Table 3: Random samples of sizes 200, 20,000 and 2,000,000 of outcomes of algorithm $\mathcal{A}$. The $slift$ and $slift_d$ measures are given with confidence intervals at significance levels of 5%.

As we can see in Table 3, all empirical fairness measures approach their true value as the sample size grows large. Furthermore, our confidence in these values also increases (results on the asymptotic normality of the empirical mutual information [40] yield a confidence interval of $[1.86 \cdot 10^{-4}, 2.43 \cdot 10^{-4}]$ at significance level 0.05 for the largest sample).

Yet, suppose that the true value of the algorithm's bias (whether measured through $\hat{I}$, $slift$ or $slift_d$), falls slightly below the policy-defined threshold appearing in Definitions 9 or 10. Then, we would consider $\mathcal{A}$ to be non-discriminatory, even when given access to large datasets in which the algorithm's inherent bias is clearly apparent. Consider the notion of a-protection from Definition 10, with a fairness threshold a of 5%, and a significance level β of 5%, $\mathcal{A}$ is considered a-protective with respect to both the $slift$ and $slift_d$ measures, if we consider our largest sampled dataset. The algorithm's bias (in the sense of $slift$ or $slift_d$) is indeed below 5%, yet our sampled data indicates that over 21,000 more men than women were hired nonetheless, which should be a clear indication of bias.

This issue is addressed in the US Uniform Guidelines On Employee Selection Procedures (UGESP), in the context of the four-fifths rule (Appendix A). The rule states that a threshold of 1.25 for the ratio of selection-rates in a hiring process should generally be considered as evidence of a discriminatory bias, but that '*smaller differences* in selection rate may nevertheless constitute adverse impact, where they are *significant in both statistical and practical terms*' and that '*greater differences* in selection rate may not constitute adverse impact where the differences are based on small numbers and are *not statistically significant*' [13].

In this sense, the defined ratio threshold of 1.25 can be understood merely as an easily understandable and usable guideline, in the context of a more general fairness notion based on the statistical significance of an observed bias. Thus, with regard to these legal guidelines, Ruggieri et al.'s approach appears to be flawed, in that it considers an algorithm as fair, even if there is overwhelming statistical evidence that the algorithm does exhibit a small bias. A similar approach is taken by Luong et al. [44]. A more appropriate formulation of Definition 10 would be to view $\mathcal{A}$ as unfair, whenever there is high confidence that the mechanism exhibits *any* bias, and thus violates the fairness notion from Definition 1.

This alternative approach to fairness, which we investigate now, consists in using statistical hypothesis tests to determine, from our sampled data, whether a given algorithm is biased or not. The definitions of threshold-fairness that we have seen so far (extended with confidence intervals), classify an algorithm as unfair, if there is a sufficient level of certainty that its bias lies above a pre-defined threshold. With statistical hypothesis tests, we will instead consider an algorithm to be unfair, whenever we have sufficient certainty (usually characterized by a *p-value*) that the algorithm exhibits any kind of bias in a systematic manner. As we will see, mutual information provides an interesting link between the two approaches, as it is directly related to a popular statistical test known as the G-test.

2.3 Towards Statistical Testing for Discrimination Discovery

Statistical tests are a common method used in legal practices to provide evidence of discriminatory behavior [25]. A comprehensive overview of the subject is given by Paetzold and Willborn in [50]. Additionally, a large body of work has focused on the specific case of testing for discrimination in hiring procedures [57, 27, 54].

In regard to fairness measures based on ratio or difference thresholds, these works highlight similar limitations to the ones we presented previously, namely that they fail to detect widespread discrimination that appears due to small biases. In contrast, statistical tests of independence have the advantage of detecting any level of bias, given that the sampled dataset is large enough. Legal scholars have argued that in traditional litigation scenarios, such as the hiring procedures of a small company, the availability of sufficient data samples may be questionable. However, we can expect *algorithmic* decision making to be applied on a much grander scale, and thus potentially producing substantial amounts of data, from which even small biases can be detected with high certainty.

2.3.1 Statistical Fairness and the G-test

The *null hypothesis* corresponding to the notion of fairness given in Definition 1 states that the algorithm is unbiased, or that O and S are statistically independent. A standard procedure to assess the validity of such an hypothesis, is to compute some test statistic and its associated *p-value*, which characterizes the probability that an unbiased algorithm would yield to discrepancies as large as those observed on the sampled data. The corresponding generic definition of fairness (for some given test statistic), that we call statistical fairness, is stated as follows.

Definition 11 (Generic Statistical Fairness)**.** *Let $S \in \mathcal{S}$ be a sensitive attribute, $\mathcal{A}$ be an algorithm with outputs $O \in \mathcal{O}$, and $(s_1, o_1), \dots, (s_N, o_N)$ be a collection of samples. Let p be the p-value obtained from some relevant test statistic for the null hypothesis $S \perp O$. Then, $\mathcal{A}$ is statistically-fair with respect to S at significance level β, if $p \leq \beta$.*

As we will see in Section 3, the statistical-testing framework allows for an extremely generic and robust notion of fairness, where various utility constraints can be expressed as alternative null hypotheses.

Many statistical independence tests have been considered in legal practices [63, 54], the most standard being Pearson's chi-squared test, Z-tests or Fisher's exact test. In this work, we focus on an alternative goodness-of-fit test known as the G-test or log-likelihood ratio test [65]. The G-test is an *approximate non-parametric* test, that is recommended for large datasets, for which exact tests are prohibitively expensive [47]. The chi-squared test is actually a second-order Taylor approximation of the G-test [33] and provides a weaker approximation to the theoretical chi-squared distribution [32]. In addition to its theoretical and practical justifications, the G-test also has an interesting link (see Proposition 13) to the fairness measure based on mutual-information that we discussed previously. To formally introduce the G-test, we need some additional notation. From the samples $(s_1, o_1), \dots, (s_N, o_N)$, we build a *contingency table* over variables S and O. For each pair $(s, o) \in \mathcal{S} \times \mathcal{O}$, we compute the frequency $f_{s,o}$ of observed samples of the form (s, o). The G-test is then given by

$$G = 2 \cdot \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} f_{s,o} \cdot \ln \left(\frac{f_{s,o}}{E_{s,o}} \right), \tag{1}$$

where $E_{s,o}$ is the expected number of observed samples (s, o) if the null-hypothesis is valid. In the case of an unbiased algorithm, we have $E_{s,o} = N \cdot \hat{P}(s)\hat{P}(o)$. If the null hypothesis is true, the distribution of G is asymptotically chi-squared, with $\mathrm{df} = (|\mathcal{S}| - 1) \cdot (|\mathcal{O}| - 1)$ degrees of freedom. The *p-value* p can then be obtained in the same way as for Pearson's chi-squared test. Specifically, $p = 1 - F(G; \mathrm{df})$, where $F(x; k)$ is the cumulative distribution function of the chi-squared distribution with k degrees of freedom. The use of the G-test leads to the following instance of statistical fairness.

Definition 12 (Statistical-Fairness)**.** *Let $S \in \mathcal{S}$ be a sensitive attribute, $\mathcal{A}$ be an algorithm with outputs $O \in \mathcal{O}$, and $(s_1, o_1), \dots, (s_N, o_N)$ be a collection of independent [3] samples from $P(S, O)$. Let p be the p-value obtained from the G-test statistic over the observed data. Then, $\mathcal{A}$ is statistically-fair with respect to S at significance level β, if $p \leq \beta$.*

Unless specified otherwise, in the remainder of this work we will use the term statistical fairness to refer to the specific fairness notion of Definition 12.

2.3.2 Relating MI-Fairness to Statistical Fairness

The following two simple results show a direct relation between fairness measures based on statistical hypothesis testing and on the notion of (non-normalized) MI-fairness.

Proposition 13. *The G-test satisfies $G = 2N \cdot \hat{I}(S; O)$.*

Proof. This is a well-known result, which we will prove here for completeness. By the definition of empirical probability, we have that $f_{s,o} = N \cdot \hat{P}(s, o)$ and by the null hypothesis of independence, $E_{s,o} = N \cdot \hat{P}(s)\hat{P}(o)$. Plugging into (1), we have

$$\begin{aligned} G &= 2 \cdot \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} N \cdot \hat{P}(s, o) \cdot \ln \left(\frac{N \cdot \hat{P}(s, o)}{N \cdot \hat{P}(s)\hat{P}(o)} \right) \\ &= 2N \cdot \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \hat{P}(s, o) \cdot \ln \left(\frac{\hat{P}(s, o)}{\hat{P}(s)\hat{P}(o)} \right) = 2N \cdot \hat{I}(S; O) \ . \end{aligned}$$

□

[3] The assumption of sample independence is necessary for many popular statistical tests, such as the G-test, Pearson's chi-squared test, or Fisher's exact test. We discuss this assumption, and mention alternative tests with weaker requirements in Section 5.3.2.

Theorem 14. *Let $S \in \mathcal{S}$ be a sensitive attribute, $\mathcal{A}$ be an algorithm with outputs $O \in \mathcal{O}$, $(s_1, o_1), \ldots, (s_N, o_N)$ be a collection of samples, and $df = (|\mathcal{S}| - 1) \cdot (|\mathcal{O}| - 1)$. Then, with respect to S, $\mathcal{A}$ is empirically $\frac{\epsilon}{2N}$-MI fair, if and only if it is statistically-fair at significance level $\beta = 1 - F(\epsilon; df)$.*

Proof. By the previous proposition, $\mathcal{A}$ is empirically $\frac{\epsilon}{2N}$-MI fair, if and only if $G \leq \epsilon$. As for a fixed k, $F(x;k)$ is a strictly increasing function, the p-value satisfies $p \leq 1 - F(\epsilon, \mathrm{df})$, if and only if $G \leq \epsilon$. □

A nice consequence of Proposition 13 and Theorem 14 is that, for a fixed sample size N, mutual information is a direct measure of the likelihood that an algorithm is biased, and thus a valid measure of fairness in the sense of Definition 1. If we recall the comparison between statistical parity and MI-fairness given in Table 2, the fact that mutual information between S and O is empirically much lower for $\mathcal{A}'$ than for $\mathcal{A}$ thus provides a direct indication that $\mathcal{A}$ is more likely to be biased. In contrast, the total-variation measure would lead to believe that $\mathcal{A}$ is the fairer of the two algorithms.

We also reconsider the situation in Table 3, reproduced here for convenience. Recall that the true mutual information between S and O is $I = 2.0 \cdot 10^{-4}$. As the sample size increases, $\hat{I}$ approaches the true value. For large datasets, we expect to have $\hat{I} \approx I$. Yet, the larger our sample is, the larger the G-test statistic becomes as well, since it grows as $N \cdot \hat{I}$. Thus, unless our algorithm is fair (which means we have $I(S;O) = 0$), the G-test statistic will naturally become significant if enough data is available. For the above samples, the G-test yields p-values of respectively 0.67, $2.47 \cdot 10^{-3}$ and $2.03 \cdot 10^{-188}$. At standard significance levels of 5% or 1%, the null hypothesis would be rejected and the algorithm classified as unfair in the last two samples, where the gender bias is clearly apparent. For the first sample, the limited amount of available data would not allow us to conclude, with high certainty, that the algorithm is indeed unfair.

	Male		Female		
$\lvert\mathcal{D}\rvert$	Applicants	Hired	Applicants	Hired	$\hat{I}$
200	100	**54.00%**	100	51.00%	$4.512 \cdot 10^{-4}$
20,000	10,015	**51.01%**	9,985	48.87%	$2.290 \cdot 10^{-4}$
2,000,000	1,000,941	**51.07%**	999,059	49.00%	$2.143 \cdot 10^{-4}$

3 Utilitarian Fairness

While the independence between algorithmic outputs and sensitive features appears to be a desirable fairness property in general, there are a number of possible scenarios where complete parity is prohibitive, because it is significantly at odds with user's or business's utility.

The considerations for user utility and business utility are rather different. When the amount of utility that a user derives from an algorithmic output depends on some sensitive feature, then an algorithm satisfying Definition 1 may still be considered unfair, because some protected class of users enjoys less benefit than another. In such a case, a biased algorithm is *desirable from the users' perspective*, if in turn the algorithm exhibits no bias on the average perceived utility.

Alternatively, the amount of utility that a business perceives for some algorithmic decision may be dependent on sensitive features, for instance if some protected attribute strongly correlates to a users' qualification with respect to the particular decision at hand. A simple example from [60] is to consider a truck company, whose hiring policy is based on applicants being in possession of a truck-driver's license. If it appears that men are more likely than women to meet the company's requirements, then a gender bias in the company's hiring procedure could be deemed as an acceptable case of *business necessity.* If the business requirement is genuine, true parity is *undesirable from the business's perspective* because it drastically impacts the business's functionality and efficiency.

3.1 User Utility

Consider an ad-placement algorithm for a shampoo brand. The cosmetic company would like to provide ads targeted to a user's hair texture, which happens to be strongly correlated to sensitive features such as gender or ethnicity. Suppose we conduct an analysis of the algorithm's decisions, in showing ads for four products A,B,C,D, over groups of users identified by gender and ethnicity. Products A and B are male shampoos, and C and D are female shampoos. In addition, products B and D are targeted for specific hair textures, that are more prominent in black people. The analysis yields the following results.

	Male		Female	
	White	Black	White	Black
A	80	20	0	0
B	20	80	0	0
C	0	0	100	0
D	0	0	0	100

Table 4: Ad placements of four shampoo products for users classified by gender and ethnicity. Products A,B are targeted towards males, and products C,D towards females. Products A,C are mainly targeted towards white users and products B,D towards black users.

The ad placement algorithm exhibits a clear bias between genders and ethnic groups. Yet, this need not be an indication of any discriminatory practices. Suppose the cosmetic company ran a survey asking users to rate their products as either 0 (not interested) or 1 (interested), with the following results.

	Male		Female	
	White	Black	White	Black
A	1	1	0	0
B	1	1	0	0
C	0	0	1	0
D	0	0	0	1

Table 5: User preferences for products A,B,C,D. Ratings are binary with values 0 (not interested) and 1 (interested).

We can interpret Table 5 as follows. Men only like products targeted towards males, but don't seem to care about particular hair types. In contrast, women only like shampoos branded as female and targeted towards their specific hair texture. Combining the results from Tables 4 and 5, it becomes clear that all users actually receive ads corresponding to their preferred product(s), regardless of any sensitive attributes. This (simplistic) example motivates a utilitarian definition of fairness, by shifting our attention from algorithmic outputs to the utility that users perceive from these outputs, and considering the correlations between this utility and sensitive features.

This example also illustrates the limitations of many fairness definitions appearing in previous works [12, 60, 58, 36, 37, 74, 22], that either implicitly or explicitly assume that algorithmic outputs can be classified as inherently positive or negative. Note that in our example (see Table 5), outputs are also classified as positive or negative but a crucial difference is that this classification is not universal, but dependent on sensitive features. Additionally, we could also consider more general classifications, where different outputs provide a range of levels of utility to users.

To characterize a user's utility, we introduce an additional random variable U over some discrete alphabet $\mathcal{U}$. Given the random variables S and O, a user's utility U is defined by a conditional mapping $P(U \mid S, O)$. Thus, the distribution of user utility for a particular algorithmic output may vary depending on the value of the sensitive attribute. Our definition of fairness can then be reformulated in terms of a user's utility as follows.

Definition 15 (User-Utilitarian Fairness)**.** *An algorithm $\mathcal{A}$ is fair with respect to a sensitive feature S and user-utility U, if and only if $U \perp S$.*

Estimating and quantifying the fairness of a given algorithm in the user-utilitarian setting relies on the same notions of MI-Fairness and Statistical Fairness as in the non-utilitarian case, except that we replace the algorithmic output O by its utility U. Thus, in addition to a dataset $\mathcal{D}$ of user attributes and the corresponding algorithmic outputs, we will also assume that we are provided with a value $u \in \mathcal{U}$ denoting each user's perceived utility from $\mathcal{A}$'s output.

In the remaining part of this work, we will make the simplifying assumption (unless specified otherwise) that the perceived utility for an algorithmic decision is the same for all users. We will thus continue reasoning about correlations between S and O rather than between S and U. Our results can be extended to the user-utilitarian case in a straightforward manner, by appropriately replacing outputs O by a user's perceived utility U.

3.2 Business Utility

The general business goal of algorithmic decision making could be defined as that of constructing and applying some matching or classification from user data to outcomes, such as to maximize the business's notion of utility. On the one hand, a business will aim to increase its utility by collecting and utilizing as much data about its users as is available. On the other hand, achieving fairness in the sense of Definition 1 requires that the algorithm's outputs be influenced only by the set of collected user attributes that are not strongly correlated to any sensitive features. When designing a fair decision making algorithm, the business thus faces a standard trade-off between fairness (which can always be achieved by using a data-oblivious algorithm) and its own utility (which can always be maximized by considering all the available data). However, this problem might fail to have any acceptable solution, in situations where some user attribute is both strongly correlated to sensitive features and indispensable in order for the business to achieve a viable utility. For instance, a business might have particular user requirements, for decisions such as hiring or credit allocation, that must be met to guarantee the business's sustainability. If a user's tendency to meet these requirements is highly dependent on some sensitive attribute, a bias in the decision making process might be considered acceptable because it is the result of a legitimate *business necessity.*

The so-called *business necessity defense* is a standard legal practice, where a business, facing accusations of discriminatory practices, provides sustainable evidence that its biased decision making is due to the necessity of users meeting certain crucial requirements [28, 56, 66]. This notion has received rather little formal treatment from the literature on fairness in algorithmic decision making so far. We review some notable exceptions in Section 4, and provide additional background on anti-discrimination legislation in Appendix A.

3.2.1 Conditional Fairness

We consider an algorithmic task for which genuine business requirements can be represented as classes (or qualification levels) $K \in \mathcal{K}$. A user's class depends on a small number of non-sensitive attributes, that are part of the user's data X. We denote these necessary features by a random variable $B \in \mathcal{B}$. Instead of having a user's class depend solely on these features, we will consider a more general setting where a user's class may also depend on other users' classifications. This will allow us to model business strategies that base decisions on a user's *relative qualification* with respect to other users.

Formally, a business provides a mapping $h : \mathcal{B}^n \to \mathcal{K}^n$, defined for any $n \geq 1$, such that an algorithm's bias is explained away, if we take into consideration the task-specific clustering of users into different classes K. We then define fairness under business necessity constraints as the *conditional independence* of S and O, given K.

Definition 16 (Business-Utilitarian Fairness)**.** *An algorithm $\mathcal{A}$ is fair with respect to a sensitive feature S and class $K \in \mathcal{K}$, if and only if*

1) There is a set of features $\mathcal{B}$ and a mapping $h : \mathcal{B}^n \to \mathcal{K}^n$, that are a valid representation of a genuine business requirement.

2) $O \perp S \mid K$.

The first condition of Definition 16 corresponds to the current legal practice of assessing the validity of a business's qualification requirements under policy. The second condition can be evaluated from empirical data, by extending our fairness measures to take into account user classes K. We will need the following additional notation. From a collected dataset $\mathcal{D} = \{(x_1, s_1), (x_2, s_2), \dots, (x_N, s_N)\}$, we extract business necessary attributes $\{b_1, b_2, \dots, b_N\}$, and compute the classes $\{k_1, k_2, \dots, k_N\}$ from the business's mapping h. Using the algorithm $\mathcal{A}$ as a black-box, we obtain N samples of the form (s_i, o_i, k_i).

We begin by introducing *conditional mutual information*, a standard measure of the conditional dependency of two random variables, given a third. Using this notion, we obtain a generalization of MI-Fairness (Definition 9), for situations with business utility constraints.

Definition 17 (Conditional Mutual Information). *For an algorithm $\mathcal{A}$, the empirical conditional mutual information between S and O, given K is defined as*

$$\hat{I}(S;O \mid K) = \mathbb{E}_K[\hat{I}(S;O) \mid K] = \sum_{k \in \mathcal{K}} \hat{P}(k) \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \hat{P}(s, o \mid k) \ln \left(\frac{\hat{P}(s, o \mid k)}{\hat{P}(s \mid k)\hat{P}(o \mid k)} \right) .$$

The normalized measure $\hat{I}_{norm}(S;O \mid K)$ is given by $\dfrac{\hat{I}(S;O \mid K)}{\min\left\{\hat{H}(S \mid K), \hat{H}(O \mid K)\right\}}$.

The generic, test-agnostic notion of statistical fairness from Definition 11 can also be extended to conditional statistical fairness, by considering the alternative null-hypothesis $S \perp O \mid K$. A particular instantiation of conditional statistical fairness, using the G-test, is obtained as follows. From our samples $(s_1, o_1, k_1), \dots, (s_N, o_N, k_N)$, we build a three-way contingency table with frequencies $f_{s,o,k}$. The null hypothesis of conditional independence yields the expected frequency $E_{s,o,k} = N \cdot \hat{P}(k)\hat{P}(s \mid k)\hat{P}(o \mid k)$. The G-test is then given by

$$G_K = 2 \cdot \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \sum_{k \in K} f_{s,o,k} \cdot \ln \left(\frac{f_{s,o,k}}{E_{s,o,k}} \right) . \tag{2}$$

If the null hypothesis is verified, G_K's distribution is asymptotically chi-squared with $(|\mathcal{S}|-1) \cdot (|\mathcal{O}|-1) \cdot |\mathcal{K}|$ degrees of freedom. Definition 12 is then extended in a straightforward manner to assess the fairness of an algorithm under business necessity constraints, by considering the test measure G_K instead of G. Furthermore, a more general form of Proposition 13 also holds in this setting.

Proposition 18. *The G-test in* (2) *satisfies* $G_K = 2N \cdot \hat{I}(S;O \mid K)$.

Proof. From (2), and using that $f_{s,o,k} = N \cdot \hat{P}(s, o, k) = N \cdot \hat{P}(k)\hat{P}(s, o \mid k)$, we get

$$\begin{aligned} G_K &= 2 \cdot \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \sum_{k \in K} N \cdot \hat{P}(k)\hat{P}(s, o \mid k) \cdot \ln \left(\frac{N \cdot \hat{P}(k)\hat{P}(s, o \mid k)}{N \cdot \hat{P}(k)\hat{P}(s \mid k)\hat{P}(o \mid k)} \right) \\ &= 2N \cdot \sum_{k \in K} \hat{P}(k) \sum_{s \in \mathcal{S}} \sum_{o \in \mathcal{O}} \hat{P}(s, o \mid k) \cdot \ln \left(\frac{\hat{P}(s, o \mid k)}{\hat{P}(s \mid k)\hat{P}(o \mid k)} \right) \\ &= 2N \cdot \mathbb{E}_K[\hat{I}(S;O) \mid K] = 2N \cdot \hat{I}(S;O \mid K) . \end{aligned}$$

□

Thus, in the presence of business necessity constraints, there is also an equivalence between MI-Fairness and statistical hypothesis testing, analogous to the result of Theorem 14.

3.2.2 Examples

To illustrate the flexibility of this framework, we now present two possible mappings h for businesses that require job applicants to hold specific educational degrees. The first mapping will correspond to a business requesting a *minimal qualification*, while the second mapping corresponds to a business asking for the *highest available qualification*, thus considering a user's relative qualification with respect to all other applicants. Both of these simple examples of business utility functions can then easily be converted into null-hypotheses, indicating what the expected fair outcomes on a particular dataset should be. As an example, we consider a business with 120 open job positions. The sensitive feature is gender, and applicants are ranked based on the their university degree (bachelors, masters or PhD) as follows.

	Male	Female
PhD	60	24
Master	240	156
Bachelor	150	270

We first consider a business that requires a Master degree as a minimal qualification. Thus, there are two qualification levels $\mathcal{K} = \{\texttt{qualified}, \texttt{non-qualified}\}$ based on the feature $B \in \{\texttt{PhD}, \texttt{Master}, \texttt{Bachelor}\}$ and $h(B) = \texttt{qualified} \iff B \in \{\texttt{PhD}, \texttt{Master}\}$. The fair null hypothesis states that 120 out of the 480 qualified applicants get hired, regardless of gender. Thus, conditioned on being qualified, 25% of both the men and women should be hired as illustrated in Table 6.

	Male		Female	
	Applicants	Hired	Applicants	Hired
PhD	60	25%	24	25%
Master	240	25%	156	25%
Bachelor	150	0%	270	0%
Total	450	**16.7%**	450	10%

Table 6: Expected fair hiring if the minimal requirement is a Masters degree.

Alternatively, the business could require the maximal available qualifications to fill in the 40 positions. The mapping h thus classifies a user as either `most-qualified`, `qualified` or `unqualified` based not-only on his own degree B, but also on the degrees of all other applicants. The fair null hypothesis, displayed in Table 7, then states that all 84 applicants with a PhD degree get hired ($K = \texttt{most-qualified}$), followed by 36 out of the 396 applicants with a Masters, regardless of gender.

For both examples in Tables 6 and 7, the hiring algorithm would be considered statistically unfair in the sense of Definition 1, because a clear dependency between gender and hiring is apparent. However, if the necessary qualifications we introduced are accepted by policy as genuine business requirements, fairness in the sense of Definition 16 is satisfied.

	Male		Female	
	Applicants	Hired	Applicants	Hired
PhD	60	100%	24	100%
Master	240	9%	156	9%
Bachelor	150	0%	270	0%
Total	450	**18.2%**	450	8.4%

Table 7: Expected fair hiring if the business ranks applicants by highest available university degree.

3.2.3 Classifier Composition

Given business-necessity requirements, expressed in terms of a mapping h from a set of features B to a class K, we can essentially view a fair algorithm as consisting of two functions, g and h, as illustrated below. Users are first clustered, based on business requirements, through the application of h. The mapping h takes as input business necessary features B, that are part of the user's data X, and outputs a class $K = f(B)$. The second function g acts only on a user's class K, as well as on the remaining non-sensitive features deemed as not necessarily essential to the classification task at hand, denoted X_{-B}, and outputs the decision O. The notion of conditional fairness introduced in Definition 16, states that the algorithm is fair, if h represent a valid business requirement, and if the outputs of g are independent of sensitive features S, for each class $k \in \mathcal{K}$. Equivalently, for each class k, g should act as a fair classifier in the sense of Definition 1.

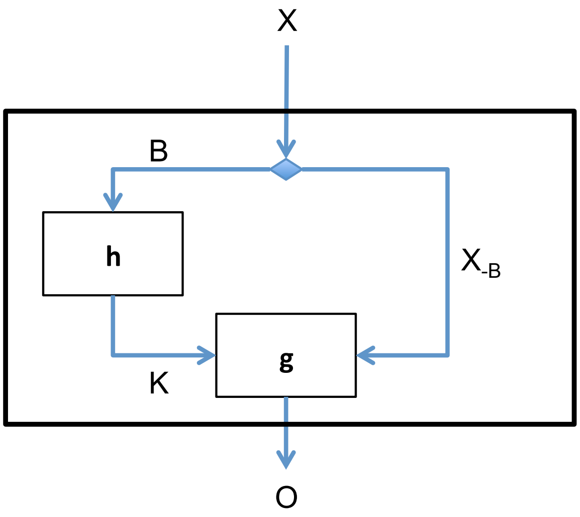

Figure 1: A fair two-stage classification of user's in the case of business necessary features. The function h maps business-essential features to a class K. Function g combines K and additional features X_{-B} to output a decision O. The mechanism is fair if O is conditionally independent of sensitive features S, given the business classification K.

4 Related Work

4.1 Fair Data Mining and Classification

The study of discrimination in data mining was instigated by Pedreschi, Ruggieri and Turini, who introduced the concept of discriminatory association rules based on lift-measures [53]. In later work [60, 58], they introduce the definitions of a-protection and propose a series of data-mining algorithms for detecting discriminatory associations.

Building upon difference- and ratio-based measures of fairness, further research from the data-mining and machine learning communities has mainly focused on the prevention of discrimination, through the construction of discrimination-aware classifiers. These works broadly fall into three different categories. The so-called *data-massaging strategy* consists in modifying data labels in the training set, such as to minimize the bias of the classifier on the test data [36, 37, 74, 44, 22]. In contrast, the *post-processing strategy* trains a classifier on the raw discriminatory data and then transforms the obtained classifier to achieve fairness [12, 30]. Finally, the *regularization strategy* adds a regularizer to the classifiers objective function, that penalizes discriminatory biases [12, 37, 38, 73]. To the best of our knowledge, the work of Kamishima et al. [38] is the only one that explicitly considers mutual information as a measure of a classifier's bias.

A different approach is taken by Dwork et al. [17] in their 'Fairness through Awareness' framework. They consider the problem of building a randomized classifier, that optimizes an arbitrary loss function under fairness constraints. They assume the existence of an arbitrary task-specific similarity metric between users. Our notion of business-utility from Definition 16 can be seen as a concrete and simple example of such a metric, where users are considered as equal for a task, if they share some necessary qualifications. As we have argued in Section 2.2.2, the fairness measures of [17, 73], that are based on either the total-variation norm (statistical parity) or the D_∞ metric, appear to be poor measures of the statistical significance of an algorithm's bias.

4.2 Information Flow Control and Fair Ad Targeting

Fairness has also been investigated in the context of information flow control, with the goal of understanding how a user's data is being transmitted and used by different entities in accordance with contextual policies [7, 70, 15]. Recently, a special focus has been given to discrimination detection in *ad targeting* [67, 42, 69, 14]. The general methodology used in these works consists in simulating a large number of fake web users, in order to measure how a triggered variation of some sensitive attribute may influence the type of ads that are displayed. As such, these methods may only detect discrimination that is *directly caused* by sensitive attributes. In contrast, works on algorithmic fairness, including our own, consider the more general problem of detecting discriminatory practices, even if they emanate from non-sensitive attributes that happen to be correlated to sensitive ones. In this context, it is essential to consider *real* user data, rather than fabricated simulations, in order to capture the true underlying correlations between multiple user attributes.

Interestingly, although they have been largely ignored in the data-mining and machine learning literature on algorithmic fairness, statistical testing methods are standard tools used in works on fairness in ad targeting. In particular, Tschantz et al. [69, 14] recently proposed a rigorous and formal methodology for reasoning about information flow experiments, by relying on a class of exact statistical tests known as permutation tests. We will discuss practical considerations related to statistical testing in more detail in Section 5.

4.3 Fairness, Privacy and Mutual Information

Many researchers have noted the intrinsic dual relationship between fairness and privacy [17, 73, 22]. From our notion of fairness (Definition 1), an algorithm is fair if its outputs are independent of the sensitive features. On the one hand, this implies that a user's sensitive features reveal nothing about the output he will obtain. This corresponds to the standard notion of parity or equity between sensitive groups. On the other hand, it also implies that the algorithm's outputs reveal nothing about the sensitive features, which can be seen as a privacy guarantee. This inherent connection between privacy and fairness is directly captured by the mutual-information measure, since $I(S; O) = I(O; S)$. Therefore, in some sense, an algorithm is only as fair, as it is private towards sensitive features. In view of this duality, some works have suggested to measure fairness in terms of the *predictability* of sensitive features given the outputs, using specific types of classifiers such as SVMs [22] or logistic regression [14]. We believe that in this context as well, mutual information provides a more general measure of fairness, as it is well known to provide unconditional *upper bounds* on any classifiers accuracy, as a consequence of Fano's inequality [21, 18]. Intuitively, the more information O provides about S, the more accurate an optimal classifier may be in predicting S. Formally, the error rate of a classifier that predicts S from O (the proportion of misclassifications of S), is lower-bounded in terms of $H(S \mid O) = H(S) - I(S; O)$, where $H(S)$ is constant.

In an early work on privacy in data-mining, Agrawal and Aggarwal proposed mutual information as a quantification of privacy loss [4]. It was later noted in [20] that mutual information only guarantees *average-case privacy* and thus may not give satisfactory protection to *outliers* with rare sensitive features, or against extremely rare privacy-breaching algorithmic outputs. In contrast, *worst-case* measures of privacy such as differential privacy [16] aim at providing unconditional protection to all users. Zemel et al. [73] have proposed a similar distinction for fairness, by considering both *group fairness*, the notion of fairness appearing in this work, and *individual fairness*, which asks that 'similar' individuals receive similar outputs. We discuss individual fairness in more detail in Section 4.5.

The general problem that Zemel et al. [73] consider is that of mapping user data to *intermediate representations*, such that the mapping preserves as much information about the data, while simultaneously hiding a binary sensitive attribute (and thus being fair). The authors note that their approach is closely related to the *information bottleneck method* [68], the aim of which is to compress some variable X, while preserving mutual information about another variable Y. As we have seen however, the statistical parity measure used in [73] is not quite equivalent to the mutual information measure appearing in the original framework. Makhdoumi et al. [45] use similar methods for the problem of mapping user data to a *privacy-preserving representation*, by minimizing the mutual information between representations and sensitive features, while maximizing the preserved information about the original data.

4.4 Utilitarian Fairness

Considerations of the utility associated with perfect equity have been remarkably absent from many previous works on algorithmic fairness. The question of user-utility, in particular, does not seem to have received any attention so far. As we already mentioned, Dwork et al. [17] consider the optimization of a generic loss function under fairness constraints, for users regarded as similar for the classification task at hand. It would thus seem that diverging user-utilities could be seemingly integrated into the optimized function. However, the fairness constraints they consider are expressed as equity constraints over the distribution of *outcomes* of the classifier, and not their associated utilities. Thus, if users with different utility criteria are regarded as similar with respect to the classification task, they must obtain similar outcomes, which is unfair to the users perceiving lower utility. If instead, these users are regarded as non-similar, then the algorithm may assign *arbitrary* outcomes to all users, and thus minimize the global loss-function without any fairness guarantees.

We now discuss previous works' considerations of business necessity, and how they relate to the definitions we introduced in Section 3.2.

The mechanism in [17] relies on a task-specific *similarity metric* for users, that could incorporate notions of business-necessary features. Contrary to current legal practices, where the burden of justifying the requirements of a task are left to the business, Dwork et al. envision this similarity metric being part of the public domain, and proposed or imposed by some external entity. Note that asking the business itself to describe a general metric is prohibitive, because assessing the fairness of such a metric might be as challenging as measuring the fairness of the decision-making algorithm in itself. However, it also seems unreasonable to assume that some external body could define a similarity metric consistent with the specific business requirements of any particular business. In our approach, instead of providing a general distance measure between all users, the business proposes a *clustering* of users into task-specific *classes* or *qualification levels*, based on only a few non-sensitive attributes. As is the case in current legal practices, such a classification will have to be evaluated under existing policies to determine whether it meets a genuine business requirement.

In their framework, Ruggieri et al. [60, 58] suggest that a business may argue against a discrimination allegation, by identifying a non-sensitive attribute, representative of a genuine business requirement, and that 'explains' most of the algorithm's bias. Our approach is comparable, although we focus on the more robust and generic fairness definitions based on statistical hypothesis testing that we introduced. Furthermore, we also consider more general business requirements than ones that can be represented by a single non-sensitive attribute.

Finally, Zliobaite et al. [74] propose to separate an observed bias into *explainable discrimination*, represented as a qualification score obtained from all the non-sensitive attributes, and *bad discrimination* based on sensitive attributes. As for many previous works on algorithmic fairness, their analysis is limited to binary sensitive features and outputs (one of which is known to be positive), as it makes use of the $slift_d$ measure. Our approach, which is conceptually similar, covers more general situations such as multivalued attributes or user-specific utilities. Instead of considering *all* non-sensitive features as potentially explanatory, we expect business requirements to be expressed using only a small number of essential features, such as to enforce transparency and justifiability of a bias in the decision-making process.

4.5 Individual Fairness

The notion of fairness we considered in this work asks that an algorithm's outputs be unbiased for specific *groups of users*, as defined by a set of sensitive features. As such, the fairness measures we considered in Section 2, including the ones from previous works, can be seen as measures of the *average-case* fairness of an algorithm, over the protected user populations. As with notions of average-case privacy, mentioned previously, measures of group fairness may not give appropriate protection to certain data outliers. For instance, consider an algorithm that generates outputs for 500,000 males and 500,000 females, with all users getting the same outputs, except for user John Doe. With respect to gender, the algorithm would be considered fair, as the algorithm's bias is too small to be deemed as significant by any of the fairness measures we considered [4]. However, unless John is somehow vastly different from the other considered users, the algorithm is absolutely unfair from his point of view.

Dwork et al. [17, 73] have proposed to distinguish between the notions of *group fairness*, asking that users from protected groups receive similar outcomes on average, and *individual fairness*, asking that users deemed as *similar*, with respect to the algorithmic task at hand, be treated similarly. A task-specific similarity metric between users is assumed to be defined and made available by policy-makers. Our notion of business-necessary features, introduced in Section 3.2, can be seen as a particular case of such a similarity metric. We will also see in Section 5.2, that *cluster analysis* can be used to learn a representation of different user sub-populations, that are regarded as different under the algorithm's classification task. A different approach is taken by Luong et al. [44], who compare a user's outcome with that of it's *k-nearest-neighbors*, thus relying on a *task-agnostic* similarity metric.

A potential issue with the approach of Dwork et al., is that they reason about individual fairness in terms of proportions of outcomes. Indeed, they consider an algorithm to be individually fair, if the distributions of outcomes for similar users are close. Imagine an algorithm for setting health-care premiums, that simply samples a premium uniformly at random, between \$0 and \$1,000,000, for each user. This algorithm is of course non-discriminatory, with respect to any sensitive feature. However, it is difficult to view it as fair to the individual, even if every user basically gets the same 'chances'. Essentially, the a priori outcomes may be equal for all users, but the a posteriori outcomes can be highly unequal. This simple example is directly related to the question 'when is a lottery fair?' [62, 11]. As noted by Sher [62], a lottery is intuitively fair, when all participants have 'equal claims to a good that cannot be divided among them'. This notion of (non)-divisibility appears to be key. For instance, if we consider a job offer with 100 equally qualified applicants, only one of them can get the job, so in this case choosing one of the applicants at random appears fair.

Perhaps, a better notion of individual fairness could be obtained by viewing a classification algorithm as a particular case of a *resource allocator*, and require that similar users receive a similar amount of resource shares, up to sub-divisibility. In the context of our health-care premium example, the individually-fair outcome would be for all users to receive the same premium, as monetary amounts are 'infinitely' sub-divisible. For the job-offer situation, the available position is not sub-divisible and should thus be fairly allocated at random over all

[4] We could also suppose that both John and Jane Doe receive some alternate treatment, in which case absolutely no gender-bias occurs.

equally-qualified applicants. There are numerous works on the fairness of resource allocation in the field of economics [5], which could serve as an inspiration for the development of a more robust definition of individual fairness, in the context of algorithmic decision making. We leave this question open for future work, and do not expand further on it here.

5 Practical Perspectives

Having discussed different fairness definitions, and having proposed a generic statistical framework for reasoning about discrimination both in the presence, or absence, of utility constraints, we now consider the problem of designing a *practical*, *robust* and *scalable* system for the detection of widespread algorithmic biases.

The main design goals we wish to address can be expressed as follows.

1. Our solution should be able to handle different types of fairness measures, with or without user- or business-utility constraints, depending on the data provided by the system's user.
2. We should not only measure the algorithm's bias over the whole user-population, but also in meaningful *sub-populations* of the users, with regard to the algorithm's classification task. The analysis of these user-clusters should help us identify either discriminatory practices occurring only in certain data niches, or alternatively potential business-necessary features.
3. The hypothesis that we test, which may provide evidence of various discrimination contexts or potential explanatory features, should have a *simple* and *easily interpretable* representation.
4. Our system should scale to large datasets and multivalued sensitive features and outputs, and rely on robust statistical testing techniques.

We introduce a simple pipeline, illustrated in Figure 2, consisting of four main processing steps: a *data collection phase*, a *cluster analysis phase*, a *interpretable hypothesis generation phase* and a *statistical testing phase*. We discuss these stages in detail further on and explain how they contribute to the expressed design goals.

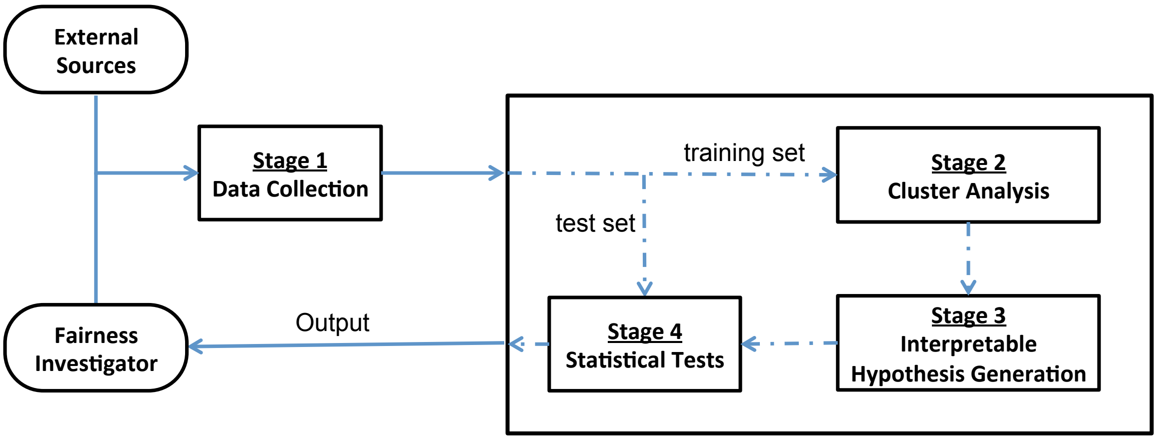

Figure 2: A generic data pipeline for statistical fairness tests.

We first introduce practical considerations regarding the collection of data in Section 5.1. In particular, we consider situations where the sensitive attribute S might not be part of the collected user data, but is inferred from some *external knowledge source*. We also also address additional steps required for cases dealing with user- or business-utility constraints, and how business-necessary features may be inferred from empirical data. In Section 5.2, we will be concerned with the difficult problem of discovering discriminatory practices arising in *subsets* of our data, while simultaneously guarding against certain famous statistical fallacies, such as Simpson's paradox. We show that data clustering can serve as a robust and scalable mechanism to analyze discrimination arising in different data niches, as well as identify potential business necessary features, the validity of which will depend on contextual policies. In particular, we will focus on decision-tree algorithms, that directly yield a simple and easily interpretable model of the underlying decision-making algorithm. Finally, in Section 5.3, we review some commonly-used statistical tools to guarantee the robustness of our fairness measures, and discuss their scalability for large datasets.

Prior work on the design of systems for measuring algorithmic fairness has been rather limited in its scope. DCUBE, proposed by Ruggieri et al. [59] appears to be the only example, to the best of our knowledge, of a system to detect discriminatory practices by analyzing *real* user data. However, as we have seen, their fairness metrics seem to be inherently limited to binary attributes, and assume *a priori* knowledge about which group is discriminated against, and which output is considered positive. Their work also doesn't consider statistical hypothesis testing methods, which form an integral part of our solution for detecting small yet widespread discriminatory practices. Alternatively, a large body of work on discrimination in ad targeting [67, 42, 14] has focused on building systems that uncover situations where a *simulated* modification of a users sensitive features triggers a significant variation in the displayed ads. Our goal is more general, in the sense that we also wish to discover algorithmic biases that are not directly caused by sensitive features, but rather by other features that are inherently correlated to sensitive attributes in the considered user population.

5.1 Data Collection, Utility and External Knowledge

For the fairness measures we introduced in Sections 2 and 3, we assume the existence of a dataset of tuples (s_i, o_i) sampled from the joint distribution of S and O. Until now, we have implicitly assumed that the sensitive features S were available as part of the collected user data X, and thus that our statistical tests could directly be applied on the available dataset. As noted by Ruggieri et al. [60], we may also wish to discover discriminatory practices against sensitive attributes that are not part of the collected user data, for instance because the collection of such data is illegal. Although the data S may not be available explicitly, it may be the case that it can be inferred through some external source of knowledge. As a concrete example, suppose we are interested in detecting discrimination against users with low salary, even though income is not part of the collected data. If our dataset contains a user's ZIP code, age or education level, we may be able to accurately infer income information from publicly available aggregate data such as that provided by the US Census Bureau [5].

Formally, we can view sources of external knowledge as an empirical joint distribution $\hat{P}(S, X)$ between sensitive attributes and collected data, that is estimated from publicly available data sources. Because S is not part of the collected data, and is thus not provided as

[5] `http://www.census.gov`

an input to the algorithm, we can make the assumption that S and O are *conditionally independent* given X (this means that two users with the same data X, but different sensitive attributes S expect to receive the same outputs). Then, we can estimate the joint probability between S and O as follows.

$$\begin{aligned}\hat{P}(s,o) &= \sum_{x\in\mathcal{X}} \hat{P}(s,o,x)\\ &= \sum_{x\in\mathcal{X}} \hat{P}(s,o \mid x)\hat{P}(x)\\ &= \sum_{x\in\mathcal{X}} \hat{P}(s \mid x)\hat{P}(o \mid x)\hat{P}(x)\\ &= \sum_{x\in\mathcal{X}} \hat{P}(s,x)\hat{P}(o \mid x)\,.\end{aligned}$$

The third equality follows from the conditional independence of S and O given X. The joint probability $\hat{P}(S,X)$ is our external knowledge, and the conditional probability $\hat{P}(O \mid X)$ can be estimated from the collected user data and algorithmic outputs. Given $\hat{P}(S,O)$, $\hat{P}(S)$ and $\hat{P}(O)$, we can estimate the mutual information between S and O and compute the G-test (or any other statistical test based on a contingency table between S and O). Concretely, a user of our system should provide either a sensitive attribute, as part of the collected data, or empirical statistics on the distribution of the sensitive attribute over the available dataset.

When reasoning about fairness with utility constraints, we further assume the presence, in the collected data, of either a utility value U (for user utility), or a qualification level K (for business utility). In Section 3.1, we modeled a user's utility for a given output by some probability distribution $P(U \mid S,O)$, capturing the fact that users with different sensitive features might derive different utilities from outputs. Here again, we will assume that this model is empirically known, meaning that $\hat{P}(U \mid S,O)$ has previously been estimated, from user studies for instance. Then, in order to evaluate the independence between S and U, we can use the fact that

$$\hat{P}(s,u) = \sum_{o\in\mathcal{O}} \hat{P}(s,u,o) = \sum_{o\in\mathcal{O}} \hat{P}(u \mid s,o)\hat{P}(s,o),$$

where $\hat{P}(s,o)$ can be estimated over the collected data.

As discussed in Section 3.2.3, in the case of business-necessity requirements, we can think of a fair algorithm as a composition of two functions (or classifiers) h and g. We know that h should depend only on a small number of essential features B, and we also assume that the function can be represented or interpreted in a simple way, since its validity should be assessed under contextual policies. We can distinguish three application scenarios for our system, depending on the information provided by the user.

- If the user can provide a representation of the mapping h and the corresponding necessary features B, the qualifications K can simply be computed for each user and added to the collected data. We are then left with the problem of assessing the conditional fairness of the classifier g, given a particular user class.

- Alternatively, the user could provide only a list of features B, that he deems essential to the classification task, but no concrete mapping h. A model for a potential function h could then be inferred from the data. For this, we can first learn an *easily interpretable* classifier for the outputs O, given the features B. For each discovered class K, we further analyze the dependence between sensitive features and algorithmic outputs. If clustering users based on business requirements K can explain any exhibited algorithmic bias, the discrimination could be justified through the necessary attributes B.

- If the user provides neither B nor h, we may still attempt to learn a simple representation of the classification algorithm, from which potential functions h and g could be inferred, depending on which features could be regarded as essential for the task at hand. We will see in Section 5.2, that *cluster analysis techniques* may lead to the *discovery* of potential business-necessary attributes, the validity of which is a matter of policy governing the particular context in which the algorithmic decision-making is applied.

For the remainder of this work, we will assume that we are given neither a complete mapping h nor essential features B. We will thus consider learning a general interpretable model of the underlying algorithm, from which potential business-necessary features could be inferred.

5.2 Subset Discrimination

When measuring the fairness of a given algorithm, one must carefully consider how the population of users u is defined, such as to detect discriminatory practices that may appear only in specific subsets of the data. An algorithm might not be significantly gender-biased in general, but only in certain *contexts*, for instance by exhibiting a bias against married women older than 50. Here martial status and age are considered to be non-sensitive features, that represent a particular *context of discrimination*, a terminology introduced by Ruggieri et al. [60, 58].

5.2.1 Simpson's Paradox

More generally, the failure to consider appropriate contexts of discrimination may lead to a situation commonly referred to as *Simpson's Paradox*, where effects that manifest in subsets of the data disappear or are reversed when considering the dataset as a whole. Examples of Simpson's paradox have been observed in various studies, notably in social sciences [39], and could lead to possibly wrong or misleading conclusions about a mechanism's (un)fairness.

Consider the following simple example, of an algorithm that decides whether a user will receive a line of credit. The sensitive attribute we consider is gender. When asking for a loan, users indicate the reason for their demand, which is part of the collected data used by the algorithm. Suppose all users are either looking to buy a new car or a new house. In the example data displayed in Table 8, the credit decision is empirically independent of the user's gender, when the full dataset is considered. However, when focusing on those users with the intent to purchase one particular item, clear gender discrimination is apparent.

Purpose	Male		Female	
	Applicants	Credit	Applicants	Credit
Buy Car	2,000	**75%**	1,000	50%
Buy House	2,000	25%	3,000	**50%**
All	4,000	**50%**	4,000	**50%**

Table 8: Discrimination in two data niches that cancel out when considering the full data. Males are favored when asking for credit to buy a car and females are favored for house purchases. Overall, men and women receive credits in equal proportions.

A famous and illustrative real-world example of Simpson's paradox is the (supposed) gender bias in UC Berkeley's graduate admissions for the fall of 1973 [9]. The overall admission figures showed that male candidates were significantly more likely to be admitted than female candidates, supposedly indicating that the admission process was discriminatory. However, when the admission results of individual departments are considered separately, there is actually evidence of a small bias in favor of women. Thus, considering each individual department as a particular data subset leads to different results than those reflected in the dataset as a whole. If it is accepted that each of the university's departments may set their own admission rates, the algorithm's bias is essentially accounted for. The apparent paradox can be explained by the fact that female candidates had a higher tendency to apply to departments with low admission rates overall.

5.2.2 Mining Discrimination Contexts

Ruggieri et al. [60, 58] refer to the problem of discovering biases in data subsets as the *inductive problem in discrimination discovery*, and introduce the concept of a *context of discrimination*, in which the fairness of the algorithm is measured. They propose a simple algorithm that mines all frequent itemsets with some minimal support from the dataset, and extracts contingency tables between sensitive features S and outputs O. The context of discrimination is then given by the remaining attributes in the itemset.

A potential issue with their approach, is that it doesn't easily scale to non-binary attributes, or large datasets. As the size of the dataset (or the number of attributes) increases, the number of itemsets with a statistically significant frequency can grow extremely large, implying that the number of hypotheses to test also explodes. The problem of multiple hypothesis testing, that we introduce in Section 5.3, becomes problematic in such a case.

5.2.3 Cluster Analysis and Decision Trees

We propose a rather different methodology compared to the one from Ruggieri et al. [60, 58], that trades *exhaustiveness* for *statistical robustness*, for the detection of biases in data subsets. Instead of considering *all* possible discrimination contexts that appear in the dataset with some minimal frequency, we will focus only on a smaller number of *significant* user subsets, unveiled by standard data clustering techniques. By applying robust statistical methods as well as corrections for multiple hypothesis testing, we will be able to guarantee that our conclusions are indeed statistically meaningful, for those user sub-populations that we uncovered.

Cluster analysis is a well-known method to deal with issues related to Simpson's paradox, by uncovering significant data niches to be considered in isolation [39]. In order to discover data subsets in which the algorithm's bias strongly deviates from the population-wide bias, we will first look to identify different sub-populations, for which the algorithm's 'behavior' differs. For instance, in the credit example in Table 8, the algorithm grants credits (regardless of gender) in $\frac{2}{3}$ of the cases to users who wish to buy a car, but only for $\frac{2}{5}$ of the users willing to buy a house. Similarly, in the Berkeley study, admission rates (again regardless of gender) vary greatly from one department to the other. Intuitively, these clusters correspond to user sub-populations, that are implicitly regarded as different under the algorithm's classification task. Identifying and analyzing these clusters can lead to the discovery of contexts of discrimination (as in the credit allocation example), or of potential business-necessity features, as in the Berkeley study. Indeed, if it is accepted that the different university departments can fix their own admission rates, then the supposed gender-bias is essentially accounted for. In the context of the framework from Section 3.2, we would say that the mapping of applicants to different classes K, based upon the particular department they applied for, represents a valid requirement that justifies the bias found in the population overall.

Note that it is always trivially possible to define user clusters in such a way that no intra-cluster bias exists, simply by grouping users based on their sensitive attributes. Alternatively, we can always exhibit some intra-cluster bias by grouping users with opposite sensitive features and outputs. This is a central issue arising in the analysis of correlations in data subsets, commonly known as the *Texas sharpshooter fallacy* [6], that we obviously wish to avoid. For this reason, we apply cluster analysis and discrimination detection in two *separate*, *independent* phases. We begin by looking for clusters of users, that are regarded as similar with respect to the algorithm's classification task, and then test, whether the algorithm exhibits a significant bias over the discovered sub-populations.

If a discriminatory bias appears in some user cluster, we are additionally interested in representing the corresponding context of discrimination in *simple interpretable terms*, in order to understand which group of users the algorithm appears to discriminate against. We thus focus on the discovery of clusters that have a simple descriptive form, using a small number of the *non-sensitive* user attributes. This step can intuitively be regarded as attempting to *learn* a simple representation or model of the black-box classification algorithm that was applied on the data.

Understanding an algorithm's decision process in a simple expressible form can help us, not only to discover discriminatory biases in specific user populations, but also to uncover explanatory attributes, that account for an observed bias. The discovery and analysis of such features may help us get a better understanding of potential business requirements, that justify an algorithm's discriminatory behavior. Indeed, if an algorithm exhibits a bias on the whole population, but this bias disappears when considering clusters defined by a small set of attributes $\mathcal{B}$, this clustering is a representation of potential business-necessary qualifications, the validity of which must be evaluated under contextual policies. If instead, some bias persists, or appears in some user cluster, this can be viewed as potential evidence that the algorithm truly discriminates based on sensitive features in this particular sub-population.

[6] http://en.wikipedia.org/wiki/Texas_sharpshooter_fallacy

Previous works that considered discrimination contexts have mainly focused on finding user sub-populations that exhibit a significant bias, and not on discovering whether a supposed discrimination over a dataset may disappear, when taking into account natural clusters of users formed by some set of explanatory features.

In regard of the above discussion, our proposed approach will consist of three generic stages. We first look to discover meaningful clusters, unveiled by the algorithm's decision process. We then form simple and interpretable hypotheses about the algorithm's bias in the different discovered sub-populations, and finally test the statistical validity of these hypotheses. We emphasize that it is important to consider the interpretability of user sub-populations *prior* to the hypothesis testing phase. Indeed, if a significant bias is discovered in some (generic) user cluster, there is no guarantee that a simple human-understandable representation of this cluster would also exhibit a statistically significant bias.

We now focus on a particular instantiation of the discussed pipeline, where cluster analysis and interpretable hypothesis generation are combined into a single step, by making use of a simple and natural class of clustering algorithms based on *decision-tree classifiers*. The main advantage of decision trees over other classification and clustering techniques is that, by design, they can yield data clusters, with respect to the classification task at hand, that are defined and easily interpretable using only a small number of features. Intuitively, a decision-tree classifier aims to represent an algorithm's *decision process*, by repeatedly splitting the data over some feature value. The leaves of the tree represent data clusters that are classified similarly. In turn, the path from the root to a leaf provides an easily interpretable encoding of the sub-population corresponding to this particular cluster. Further advantages of decision-tree algorithms, over common methods such as *K-Means*, *Mixture Models* or *Dimensionality Reduction*, include their ability to handle both numerical and categorical attributes, as well as multivalued outputs in a rather straightforward manner. In turn, a known weakness of decision-trees is their tendency to *over-fit*, by creating overly-complex and unstable decision structures. We discuss some protections against over-fitting, that directly follow from our design goals, further on.

We now discuss how decision-tree clustering can be incorporated into our data pipeline. We note that other techniques, that also yield simple representations of user sub-populations, could alternatively be considered. We discuss some potential candidates in Section 6.5.

We begin by splitting the available data into a training set and a test set. The training set will be used to identify meaningful clusters, and the statistical tests will be applied to the test set, so as to be independent of the model selection. The sensitive features are removed from the training set, as we do not want users to be further clustered based on these features. To guard against over-fitting, and to ensure that the identified clusters only depend on a small number of non-sensitive features, we bound the height of the learned tree model to some small value. We further set a threshold for the minimal size of a leaf cluster, again as a standard defense against over-fitting, but also to preserve some meaningfulness for our statistical tests [7]. These parameters could either be fixed, user-defined, or selected through

[7] We note that more complex methods, such as *boosting* and *random forests*, could be applied to obtain more robust decision trees, but at the expense of the simple interpretability of the clusters we discover. We discuss possible extensions of our model, by making use of generic *feature selection* methods in Section 6.5.

cross-validation, in which case we further need to extract a validation set from the training data. Finally, statistical hypothesis tests are applied both to the full test set, as well as to the identified clusters (also on the test set). Note that we are potentially performing *multiple* and *non-independent* statistical tests, and must therefore adjust our conclusions accordingly, as we discuss in more detail in Section 5.3. The results of our tests, as well as the learned tree-model, are output and presented to the system's user. Particular conclusions concerning the validity of the identified discriminatory practices, or business-necessary features, will depend on context and on policies governing the particular algorithmic task and sensitive features being considered.

We mention that our approach differs from the one of Kamiran et al. [37], whose goal is to learn a non-discriminatory decision tree, in order to obtain a fair classifier. In contrast, we use decision trees as a means of finding meaningful and interpretable user clusters, in which we then test for discriminatory biases.

Decision-trees also appear in the work of Luong et al. [44], with the similar goal of finding interpretable descriptions of discriminated subsets. The main difference is that they first discover discriminated *individuals*, using a form of k-nearest neighbor classification, and then train a decision tree solely to obtain a simple interpretation of the discriminated sub-population. A subtle but important issue with their approach is that they first look for sub-groups exhibiting discrimination, and then attempt to interpret these results using decision-tree classifiers. This could result in potentially misleading conclusions, because the significant discrimination exhibited in some sub-groups does not necessarily persist in the derived simple interpretations of these sub-groups. This is a reason why our focus is first on the discovery of simple interpretations of user clusters, followed by discrimination detection in the sub-populations defined in these simple terms.

5.3 Robust Statistics

5.3.1 Estimators and Approximations

While theoretically sound, many statistical methods are based on results, that hold only in the asymptotic range as the size of the dataset grows to infinity. For instance, the G-test we introduced in Section 2.3 is known to asymptotically approach a chi-squared distribution, if the hypothesis of independence holds. In practice, the inherent 'finiteness' of our datasets must be taken into account, in order to assess the validity of our statistical conclusions. Furthermore, we must always consider the assumptions about the data that different statistical methods imply, and how these translate to our setting.

We begin by considering our simple 'threshold' fairness measure based on mutual information from Definition 9. Note that given a finite dataset, computing the empirical mutual information as given in Definition 8 yields a *biased estimator* of the true value $I(S;O)$ (the bias of an estimator $\hat{\theta}$ of some real value θ is $\mathbb{E}_\theta[\hat{\theta} - \theta]$). It is actually known that for the discrete case, *no unbiased estimator exists* for general distributions [51, Proposition 8]. However, even when they exist, unbiased estimators are not necessarily optimal with respect to other loss functions such as the mean squared error, $\mathrm{MSE}(\hat{\theta}) = \mathbb{E}_\theta[(\hat{\theta} - \theta)^2]$. The empirical estimation of entropy and mutual information is an extremely important topic in information theory, and a variety of (biased) estimators improving upon the trivial estimator from

Definition 8 are known [51, 41].

In any case, we have argued that threshold-based measures of fairness are insufficient, if our goal is to assess whether an algorithm exhibits any kind of bias on a large scale. In such a case, statistical hypothesis testing methods effectively allow us to compute the probability (in the form of a p-value) that an unbiased algorithm would have produced results as extreme as the ones we have observed. This probability can be computed directly, using a so-called *exact test* such as Fisher's exact test or a permutation test [49, 69, 14]. However, these tests become computationally expensive for large datasets. Approximate tests such as Pearson's chi-squared test or the G-test have the advantage of being efficiently computable, while also yielding good approximations of the exact p-values, for large datasets. Because these approximations tend to underestimate the true p-values, thus leading to a higher rejection rate of the null-hypothesis, certain heuristic *corrections* can be applied, especially for small datasets. The most common methods are Yates' correction for continuity and Williams' correction [65, 47]. In his popular 'Handbook of Biological Statistics' [47], McDonald recommends exact tests for datasets of size up to 1000, and G-tests or chi-squared tests for significantly larger datasets.

5.3.2 Independence Assumptions

A crucial assumption required for hypothesis tests such as the G-test, chi-squared test, or Fisher's exact test, is that, under the null-hypothesis, the data samples (s, o) are *independent* samples distributed as $P(S, O)$. In contrast, permutation tests are based on the strictly weaker assumption of *exchangeability*, meaning that any permutation of the data should be equally likely, if the null-hypothesis holds. As noted by Tschantz et al. [69, 14], permutation tests may be preferable in contexts such as ad targeting, where the assumption of independence can be unreasonable. For large datasets however, permutation tests are prohibitively expensive, in which case *approximate permutation tests*, also called *Monte Carlo tests*, can be used. These tests consider a fixed number of randomly chosen permutations and yield an estimated p-value $\hat{p}$ along with a *confidence interval* for the true p-value. In this work, we have assumed that the assumption of independence of our data samples holds, and we will therefore not go into more detail about such tests.

5.3.3 Multiple Hypothesis Testing

Tschantz et al. [69, 14] also noted that many recent works on algorithmic fairness, that rely on statistical testing methods, fail to account and correct for *multiple hypothesis testing.* When considering *subset discrimination* as in Section 5.2, we are potentially interested in computing multiple hypothesis tests simultaneously, to uncover discriminatory biases in various subsets of the user population. As the number of tested hypotheses increases, so does the probability of obtaining a low p-value strictly by chance. Suppose we test a single hypothesis H, and reject it if the p-value falls below some threshold, say 0.05. Then the probability of incorrectly rejecting H (a false-positive or Type I error) is 5%. If instead we test m hypotheses $H_1, H_2, \ldots, H_m$, the probability that any of the hypotheses is falsely rejected is much larger (up to $5m\%$ if the tested hypotheses are not independent). In our case, a false-positive corresponds to mistakenly classifying a fair algorithm as unfair for some subset of the users. Because of the possibly far-reaching consequences of such an error, we should aim to limit their probability of appearance to some small value, regardless of the number of tested hypotheses.

Many statistical methods exist, that *adjust* p-values to account for multiple hypothesis testing. A common technique consists in controlling the so-called *familywise error rate* (FWER), by guaranteeing that the probability of *any* false-positive, over all tested hypotheses, falls below some pre-defined threshold (e.g. 0.05). The Šidák-correction (for independent tests only) as well as the Bonferroni and Holm-Bonferroni corrections [64, 34, 61, 3] are the simplest examples of such methods. Although they provide very strong bounds on the false-positive rate, these techniques are sometimes considered to be overly conservative, and to have limited statistical power to detect any significant biases, especially when the number of tested hypotheses is large. A less conservative approach is to consider the *false discovery rate* (FDR) rather than the FWER, which is defined as the expected number of false-positives over all rejected hypotheses. Bounding the FDR at 0.05 means that at most 5% of the rejected hypotheses are expected to be wrongly rejected. The Benjamini-Yekutieli procedure is a common method used to adjust p-values in order to control the FDR, for both independent or dependent families of hypotheses [8]. We note that adjustments for multiple hypothesis testing should also be applied when computing *confidence intervals* [61], a problem which was not addressed in the works of Ruggieri et al. [60, 58] and Luong et al. [44].

6 Evaluation

In this Section, we present an evaluation of a prototype of the system we described in Section 5. We focus on three different datasets, described hereafter. The source code and data we used can be obtained from the following `Git` repository: `https://git.epfl.ch/repo/algo-fairness.git`.

Dataset 1: Toy Credit Allocation The first dataset we use is an artificial toy dataset, extending the example from Table 8, illustrating Simpson's paradox. The data represents credit allocations to users (either male or female) for a particular purchase (a car, a house or a trip).

Purpose	Male		Female	
	Applicants	Credit	Applicants	Credit
Buy Car	2,000	**75%**	1,000	50%
Buy House	2,000	25%	3,000	**50%**
Buy Trip	2,000	**50%**	2,000	**50%**
All	3,000	**50%**	6,000	**50%**

Table 9: Toy dataset of credit allocations.

Removing the sensitive attribute (gender), this dataset contains 3 clusters corresponding to the different types of user purchases. In the dataset as a whole, as well as in the cluster of users willing to buy a trip, gender and credit allocation are independent. For users wishing to buy a car or a house, the credit allocation algorithm is biased respectively against females and against males.

Dataset 2: Berkeley Admissions Our second dataset consists of the admission figures for the six largest departments at UC Berkeley in the fall of 1973 [9, 23]. This is a classical example of Simpson's paradox, where the supposed gender-bias observed over the whole dataset disappears (or is even reversed) in each subset of the data.

Department	Male		Female	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	**82%**
B	560	63%	25	**68%**
C	325	**37%**	593	34%
D	417	33%	375	**35%**
E	191	**28%**	393	24%
F	272	6%	341	**7%**
All	2,590	**46%**	1,835	30%

Table 10: Berkeley admissions data for the fall of 1973 [9, 23].

This dataset is interesting to us for multiple reasons. It is an illustration of a potential case of business-necessity, as most of the observed discrimination can be 'explained' by clustering users into the respective departments they applied to. In the departments themselves, we can also differentiate between apparent biases that are statistically significant or not. At a standard significance level of 0.05, only the bias in department A (which is actually in favor of women) is significant enough, for the hypothesis of fairness to be rejected.

Dataset 3: US Adult Census The last dataset that we use is the famous US census data [8], for predicting whether an individual earns over $50,000 a year. The dataset consists of 48,842 instances, represented by 14 features (6 continuous, 8 categorical) in addition to the target feature. The sensitive feature we will consider is gender. In the complete dataset, a significant bias against women appears.

Male		Female	
Applicants	> 50K	Applicants	> 50K
32,650	**30.38%**	16,192	10.93%

Table 11: Gender bias in the complete Adult dataset.

We will be interested in analyzing clusters representing different sub-populations of the users, and find out whether the observed bias can be explained by some of the available features, or if it appears to be inherently present. We note that this data does not correspond to an actual algorithmic decision making process, but is nevertheless interesting to consider because of its thorough usage and analysis in the literature.

6.1 System Setup and Requirements

We built a prototype of our system in Python (version 2.7.6). All experiments were run using the IPython kernel [55] (version 2.3.1), making use of the standard data processing and machine learning libraries detailed below.

Library	Version	Purpose
numpy	1.9.0rc1	Multidimensional data processing [72].
pandas	0.16.1	Multidimensional data processing [48].
pydot	1.0.29	Graphical output of decision trees.
scikit-learn	0.16.1	Machine learning routines [52].
scipy	0.13.0b1	Statistical tests [35, 72].
statsmodels	0.6.1	Multiple hypothesis testing.

6.2 Methods

For the three datasets, we apply the generic pipeline for the robust statistical detection of subset-discrimination, as described in Section 6. Since we are performing statistical hypothesis tests on the testing set, we should make sure that the size of the test set is sufficient to guarantee meaningful statistical results. Compared to a traditional classification task, we will thus aim to learn our model on a rather small training set (yet sufficiently large to find

[8] `https://archive.ics.uci.edu/ml/datasets/Adult`

meaningful clusters), and keep most of the data for the hypothesis testing phase. In our experiments, we split our data into a training and test set using a random ($\frac{1}{4}$, $\frac{3}{4}$) split.

We train a decision-tree classifier on the training set, specifying a maximal tree height and a minimal leaf size. This is both to guard against over-fitting and to maintain the meaningfulness of our statistical tests. To assess the performance of our decision tree, we also train a baseline logistic-regression classifier. We report the accuracy of both classifiers on the test set. For the test set as a whole, as well as for each cluster unveiled by the decision tree, we perform a G-test of independence and compute the corresponding p-value. We correct for multiple hypothesis testing using the robust, yet-conservative, Holm-Bonferroni method, for a family-wise error rate of 5%.

Algorithms We use the `DecisionTreeClassifier` class from the `sklearn.tree` module in scikit-learn. The underlying algorithm used is an optimized version of CART [10], which constructs a binary tree by recursively selecting the splitting feature that yields the largest information gain for classification. We fix the decision tree's maximal height (or depth) to either 4 or 5, and the minimal leaf size to 100, guaranteeing that the discovered clusters (on the training set) contain at least 100 individuals and can be described using at most 4 to 5 non-sensitive features and thresholds. As a baseline, we use the `LogisticRegression` classifier from scikit-learn's `sklearn.linear_model` module.

Data Pre-Processing For each dataset, we use *one-hot-encoding* to transform categorical features into a set of binary features. A nice property of decision-tree classifiers is that they can directly handle both discrete and continuous numerical features with different ranges, without any normalization, and therefore do not necessarily require any additional pre-processing. For the logistic regression classifier, we first apply a standard normalization to our data to obtain centered features with unit variance. We also remove the sensitive features from the training set before learning our classifiers, since by definition we do not want to consider a clustering of users based on these values.

6.3 Results

The decision trees produced by our classifiers can be visualized using the Graphviz package [9], and we display them in Figures 3-5. Internal nodes contain a *decision rule*, which is simply a predicate involving a feature and a chosen threshold. The left sub-tree of a node corresponds to the subset of data samples for which the predicate is true. All nodes further contain the total number of samples rooted at that node, as well as the entropy of the distribution of target values. Each leaf displays the frequencies of the target values in the corresponding user-cluster.

Toy Credit Dataset The decision-tree learned on the training set is displayed in Figure 3. The tree classifier is easily seen to be optimal, given that only one feature is available. It achieves the same accuracy of 58.18%, on the test set, as the baseline logistic regression classifier. The results of the hypothesis testing phase on the test set are displayed in Table 12.

[9] `http://www.graphviz.org/`

As expected, no significant bias is uncovered when considering the whole dataset, or those users willing to buy a trip. In the two remaining clusters however, a clear discrimination emerges, once in favor of men and once in favor of women.

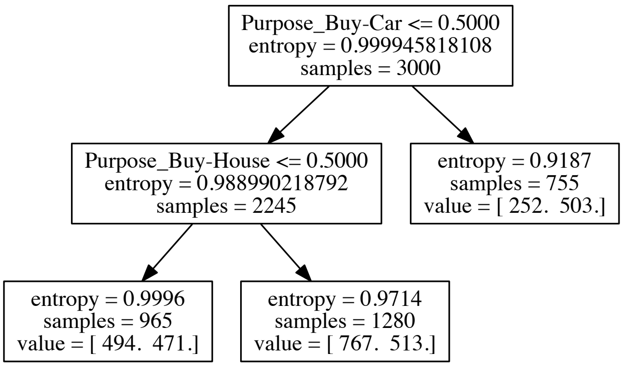

Figure 3: Decision-tree classifier trained on the toy credit dataset.

	Purpose				
	Buy-Car	Buy-House	Buy-Trip	p-value	adj. p-value
ROOT	-	-	-	0.75	1.00
LEAF 1	-	-	✓	0.68	1.00
LEAF 2	-	✓	-	$\mathbf{1.33 \cdot 10^{-56}}$	$\mathbf{5.30 \cdot 10^{-56}}$
LEAF 3	✓	-	-	$\mathbf{5.64 \cdot 10^{-32}}$	$\mathbf{1.69 \cdot 10^{-31}}$

	Male		Female	
	Applicants	Credit	Applicants	Credit
LEAF 2	1,479	24.54%	2,241	**50.16%**
LEAF 3	1,501	**75.02%**	744	49.87%

Table 12: Hypothesis testing on the full test-set, as well as on the discovered clusters for the toy credit dataset. We report the individual p-values from the G-test, as well as the adjusted p-values after applying the Holm-Bonferroni correction. For those data subsets exhibiting a bias significant at a level of 5% (displayed in bold), we further provide a detailed view showing the *direction* of the bias.

Berkeley Dataset We display the decision tree from the Berkeley admissions data in Figure 4. As for the toy credit dataset, the decision-tree classifier is trivially optimal, achieving the same accuracy of 69.78% as the baseline logistic regression. The results of the hypothesis testing phase on the test set are displayed in Table 13. We obtain a clear representation of Simpson's paradox, with the overall bias disappearing, or being reversed (in department A), when considering subsets of the population.

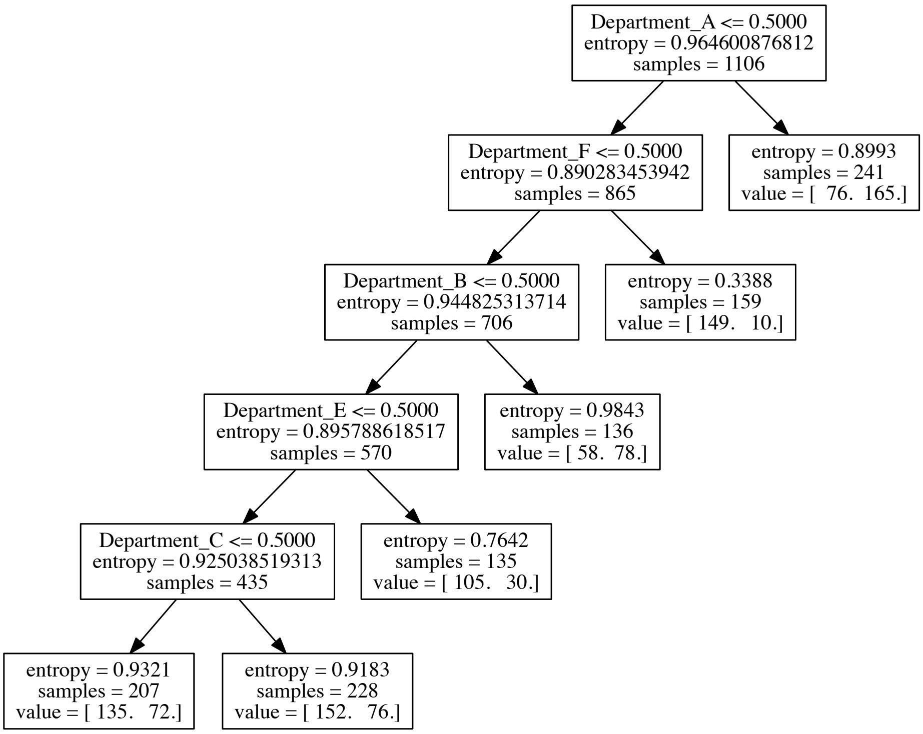

Figure 4: Decision-tree classifier trained on the Berkeley admissions dataset.

	Department							
	A	B	C	D	E	F	p-value	adj. p-value
ROOT	-	-	-	-	-	-	$\mathbf{9.23 \cdot 10^{-19}}$	$\mathbf{6.46 \cdot 10^{-18}}$
LEAF 1	-	-	-	✓	-	-	0.22	1.00
LEAF 2	-	-	✓	-	-	-	0.30	1.00
LEAF 3	-	-	-	-	✓	-	0.66	1.00
LEAF 4	-	✓	-	-	-	-	0.43	1.00
LEAF 5	-	-	-	-	-	✓	0.52	1.00
LEAF 6	✓	-	-	-	-	-	$\mathbf{8.40 \cdot 10^{-05}}$	$\mathbf{5.04 \cdot 10^{-04}}$

	Male		Female	
	Applicants	Admitted	Applicants	Admitted
ROOT	1,944	**45.99%**	1,375	30.84%
LEAF 6	617	60.62%	75	**82.67%**

Table 13: Hypothesis testing on the full test-set, as well as on the discovered clusters for the Berkeley admissions dataset.

Adult Dataset Finally, we train a decision tree on the adult census data and display it in Figure 5. We limit the tree depth to 4, to avoid an explosion in the number of leaves, for no significant gain in accuracy. The decision tree has an accuracy of 84.39% on the test set, slightly less than the 84.91% of the logistic regression classifier. The hypothesis testing phase results appear in Table 14. The features *Age*, *Capital Gain*, *Capital Loss* and *Hours per week* are continuous and self-explanatory. *Education-Num* is an integer between 1 and 16, representing a user's education level. Finally, *Married-civ-spouse* is a binary feature indicating whether a user is married or not.

	Features										
	Age > 27	Age > 55	Capital Gain > 5095	Capital Gain > 7055	Capital Loss > 1782	Education-Num > 12	Education-Num > 8	Hours per week > 44	Married-civ-spouse > 0	p-value	adj. p-value
ROOT	-	-	-	-	-	-	-	-	-	**0.00**	**0.00**
LEAF 1	-	-	-	✗	-	✗	-	✗	✗	$\mathbf{2.46 \cdot 10^{-05}}$	$\mathbf{2.46 \cdot 10^{-04}}$
LEAF 2	-	-	-	✗	-	✗	-	✓	✗	**0.04**	0.28
LEAF 3	✗	-	-	✗	-	✓	-	-	✗	0.72	1.00
LEAF 4	✓	-	-	✗	-	✓	-	-	✗	$\mathbf{9.09 \cdot 10^{-10}}$	$\mathbf{1.00 \cdot 10^{-08}}$
LEAF 5	-	-	-	✓	-	-	-	-	✗	$\mathbf{4.58 \cdot 10^{-03}}$	**0.04**
LEAF 6	-	-	✗	-	-	✗	✗	-	✓	0.16	1.00
LEAF 7	-	-	✗	-	-	✗	✓	-	✓	0.58	1.00
LEAF 8	-	-	✗	-	✗	✓	-	-	✓	0.47	1.00
LEAF 9	-	-	✗	-	✓	✓	-	-	✓	0.14	1.00
LEAF 10	-	✗	✓	-	-	-	-	-	✓	0.48	1.00
LEAF 11	-	✓	✓	-	-	-	-	-	✓	0.44	1.00

	Male		Female	
	Applicants	Credit	Applicants	Credit
ROOT	24,535	**30.17%**	12,097	10.99%
LEAF 1	5,904	**2.17%**	7,199	1.22%
LEAF 4	1,308	**24.85%**	1,461	15.54%
LEAF 5	218	**98.17%**	134	91.79%

Table 14: Hypothesis testing on the full test-set, as well as on the discovered clusters for the Adult Census dataset.

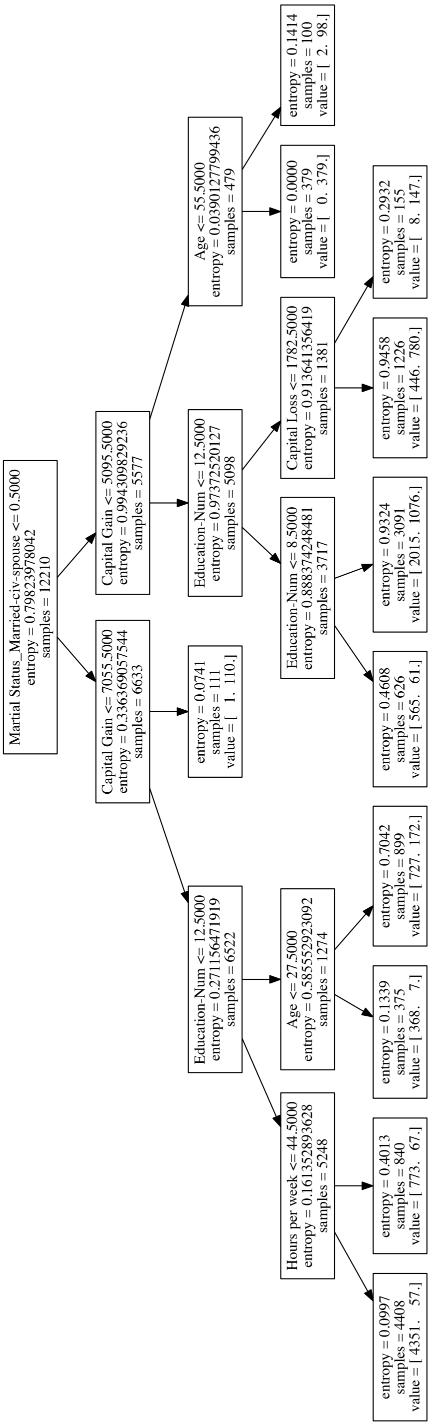

Figure 5: Decision-tree classifier trained on the Adult Census dataset.

6.4 Analysis

The credit dataset (Table 9) and Berkeley datasets (Table 10) were used because they provide clear illustrations of Simpson's paradox. As we have seen, decision-tree classifiers yield a clear characterization of the different user sub-populations for these simple examples. For the toy credit dataset, as expected, the cluster analysis and hypothesis testing phase uncovers extremely significant biases in the sub-clusters corresponding to users willing to purchase either a car or a house. For the Berkeley admissions data, the discovered clusters trivially correspond to the different departments, of which only the first exhibits a statistically significant bias, albeit in favor of women. Leaving department A aside, our results show that clustering users based on the department they apply to can be seen as potential business requirement, which justifies the bias observed over the university as a whole.

Formally, we can compute the *conditional fairness* of the admission process, as described in Section 3.2. The mapping h trivially classifies applicants based on the department they applied to (we have $\mathcal{B} = \mathcal{K} = \{A, B, C, D, E, F\}$). The G-test of conditional independence, evaluated on the full test-set, yields a p-value of 0.0037. Thus, we see that when conditioning on the department, the statistical significance of the gender-bias remains significant, but is greatly reduced, when compared to the p-value of $9.23 \cdot 10^{-19}$ obtained for the the hypothesis of unconditional independence on the full test-set.

Our results on the Adult Census dataset demonstrate the scalability of our approach, both in terms of the dataset size and in the number of features. An analysis of the features selected in the decision tree training process sheds light on some potential explanations for the observed gender bias. Intuitively, one would expect attributes such as *Age*, *Capital Gain*, *Capital Loss*, *Education-Num* and *Hours per week* to be strongly correlated to a user's income level, regardless of gender. If all of the observed gender bias could be explained using only these features, we would have potentially uncovered a set of explanatory features. The remaining feature, *Married-civ-spouse*, which indicates whether a user is married or not, seems less intuitive. First of all, our decision tree discovered that it is the feature with the highest explanatory power towards a user's income level. Over the full dataset, 44.61% of the 22,379 married users have an income higher than 50K, against only 6.44% of the 26,463 unmarried users. A further striking observation, from Table 14, is that the gender bias essentially disappears when considering only married users. We believe that a simple, and somewhat perverse, explanation of this social phenomenon, lies in the fact that a married user's income would consist of the *household income*. Thus, on the one hand, married users would be more likely to have high income, because they often combine two incomes. On the other hand, men and women would trivially appear to be equally likely to have a high household income, even if on average married women possibly still earn less than married men.

As the available data doesn't allow us to confirm or infirm this supposition, we now shift our focus to unmarried users, for which a clear gender-bias persists. Out of the decision tree's clusters, three exhibit a statistically significant gender bias, after correction for multiple hypothesis testing. For instance, we can conclude that for unmarried users with high capital gain (over 7,055), men remain significantly more likely to have high income than women. The same observation can be made for unmarried users with low capital gain, education number and hours of work per week (LEAF 1), or unmarried users older than 27, with high education number but low capital gain. In contrast, very young users (LEAF 3) seem to globally have similarly low incomes, regardless of gender.

Although the bias considered here is not induced by algorithmic decision making, but rather a striking example of *social discrimination*, the discrimination-discovery pipeline we introduced in Section 5 remains highly relevant. The use of decision-tree clustering techniques has enabled us to uncover various subsets of the data, in which vastly different discriminatory biases occur. As illustrated by our discussion on the influence of a user's martial status in the Adult dataset, or of an applicant's department in the Berkeley dataset, understanding whether an explanatory attribute should be considered as demonstrative of a valid business necessity remains a difficult question, governed by contextual policy.

6.5 Possible Limitations and Extensions

Learning a Xor with a Tree As we mentioned in Section 5.2, our approach to subset-discrimination, based on decision-tree classification, trades the exhaustiveness of the discrimination discovery, for the interpretability of the discovered subsets, as well as the meaningfulness of the multiple statistical test we perform. As a result of the decision tree construction, we might fail to find discriminatory biases in data niches that provide low explanatory power about the output feature. Consider the following situation for instance.

Purpose	Male		Female	
	Applicants	Credit	Applicants	Credit
Buy Car	1,000	**75%**	1,000	25%
Buy House	1,000	25%	1,000	**75%**
All	2,000	**50%**	2,000	**50%**

Table 15: Illustration of issues with feature selection for decision trees.

To select which split to apply to a node, standard decision-tree classifiers consider the *information-gain*, that each feature provides about the output variable. In the above example, the original entropy of the output is 1 bit, and no-single feature provides any information gain. Such 'XOR'-like scenarios are known to be difficult to learn using decision tree classifiers, which might be a limitation of our approach. However, it is unclear to us whether such 'worst-case' situations, where different subsets exhibit completely opposite discriminatory biases, should be expected to arise in the analysis of discriminatory practices.

Constraining the Tree Structure: Over-Fitting and Interpretability Another aspect of our model, that merits discussion, is the choice of constraints that we impose on the decision-tree structure, in order to protect against over-fitting while simultaneously guaranteeing interpretability for the clusters. Simply bounding the tree depth, as in Section 6, is probably not always the best possible approach. For instance, for datasets such as the Berkeley admissions, with a highly explanatory categorical feature, the resulting tree is degenerate, with one leaf per value taken by this feature. When such a categorical feature takes a large number of values (suppose we considered the 20 largest departments at UC Berkeley), the optimal tree has high depth, but each leaf is still defined in terms of a single feature. A simple solution would be to consider more general trees than the binary trees implemented in the scikit-learn library, by allowing multi-way splits on categorical features. Alternatively, we could also consider bounding the *number of leaves* of the decision-tree, rather than its depth. The most optimal approach would be to design a specific *node-splitting criterion* or *pruning strategy* for a generic decision-tree classifier, taking into account the *interpretability* of the tree nodes (the number of different features required to define the leaf clusters) as well as the leaf-cluster size (to guarantee statistically meaningful results).

Feature Selection and Small Conjunctive Formulas Finally, we note that a decision-tree classifier can be seen as an implicit method for *feature selection*. Indeed, most decision tree algorithms proceed by recursively looking for the best possible feature to split on (based on some specific performance measure). We could consider the following, different but conceptually similar, method for uncovering user sub-populations, relying on a generic feature-selection procedure. For simplicity, assume all features are binary and that we select k features $\{f_1, f_2, \ldots, f_k\}$. Then, potentially significant user-subsets can be formed by considering all possible *conjunctive formulas* over these features. For instance, if the relevant features are $\mathit{Age} \in \{\mathit{young, old}\}$, $\mathit{Capital} \in \{\mathit{low, high}\}$ and $\mathit{Education} \in \{\mathit{low, high}\}$, we would consider the 8 user sub-sets of the form $\{\mathit{Age} = x \wedge \mathit{Capital} = y \wedge \mathit{Education} = z\}$. These clusters can actually be viewed as the leaves of a decision tree, that splits the dataset based on one of the relevant features at each stage. By appropriately bounding the number of selected features k, we obtain clusters that are interpretable in terms of at most k features, and we limit the number of tested hypotheses to 2^k. A similar approach was recently introduced in the context of discrimination discovery in ad targeting [26], with sparse linear regression being used as a feature selection procedure. Feature selection is an extremely important topic in machine learning, and a variety of alternative techniques are known [29], which could also be considered.

7 Conclusion and Future Work

Results and Contributions In this Thesis, we have revisited the problem of discrimination discovery in algorithmic decision making, and proposed a statistically-robust framework, along with new metrics and approaches, for reasoning about algorithmic fairness.

We first presented an overview of previously considered 'threshold'-measures of an algorithm's bias, highlighting their inherent limitations, such as their non-applicability to multivalued features or decision processes, as well as their failure to encompass various notions of utility. We introduced a more general and scalable fairness measure based on mutual information, that had received little attention so far, and showed that it produced much more coherent results than the statistical parity measure proposed by Dwork et al. [17, 73], when applied to empirical data.

We further illustrated the awkwardness of all the considered threshold-based fairness measures, in their inability to detect small algorithmic biases applied on a large scale. We argued that the a-protection approach of Ruggieri et al. [60, 58], wherein an algorithm is considered unfair whenever there is strong evidence that its bias lies above some fixed threshold, is not in accordance with certain legal guidelines, such as those of the EEOC in the US. We introduced a generic measure of fairness based on statistical hypothesis testing, that albeit being a standard tool in legal practices, had not yet been considered in works on algorithmic fairness in the data mining and machine learning communities. In this context, we also presented another advantage of mutual information over other previously considered measures of fairness, in that it is directly related to a popular statistical goodness-of-fit test, known as the G-test.

We further discussed possible generalizations of our statistical framework, in order to account for situations where complete independence between sensitive features and an algorithm's outputs might be unacceptably at odds with the utility derived by the algorithm's vendor or its users. We first introduced a straightforward distinction between an algorithm's output and the utility that a user may derive from it, which, despite its simplicity, has not been mentioned in any of the related literature. We further considered discriminatory biases, that are deemed as genuinely necessary in the context of the algorithm's classification task. We proposed a notion of conditional fairness, wherein an algorithm's bias is measured only with respect to classes of users regarded as equally qualified by the algorithm's vendor. Such a user clustering can be seen as a particular instance of a task-specific user-similarity metric, introduced by Dwork et al. in their framework [17]. Our method closely mimics legal practices, in that it may leave the task to the algorithm's vendor, of providing a reasonable and comprehensible explanation of the business requirements that lead to a perceived bias, the validity of which must be assessed by policy.

In a second part of this work, we focused on practical aspects related to the design of a statistically-robust methodology for the discovery of discriminatory patterns in data. We first discussed various issues relating to the collection and representation of data, including aggregate statistics obtained from external sources and knowledge. In a second step, we considered the important problem of discovering discriminatory biases, that only appear in particular subsets of the user-space, forming so-called contexts of discrimination [60, 58]. We proposed a generic data processing pipeline, relying on cluster analysis techniques, to learn a simple and interpretable model of the underlying classifier, as well as the main user sub-populations with regard to a specific task. We argued that by forming simple and interpretable representations of these user clusters, we could hope to discover both specific discrimination contexts, as well as potential explanatory features for an algorithm's bias. Throughout the design of our system, we encountered various statistical and logical pitfalls, related to Simpson's paradox and the multiple comparisons problem, and proposed robust statistical methodologies to overcome these fallacies.

We presented a simple instantiation of our data pipeline, making use of decision-tree classifiers to obtain simple and easily comprehensible interpretations of the main user sub-populations in a dataset. We ran various experiments of our solution, on a toy credit-allocation dataset, the Berkeley graduate admissions data as well as the US Census dataset. For the toy dataset, we illustrated how clustering techniques may unveil significant discriminatory practices arising in particular data niches, but hidden in the complete dataset. In the Berkeley dataset, we considered an alternative example of Simpson's paradox, where a perceived bias is explained away, or even reversed, when the data is partitioned into meaningful subsets. Finally, on the Census dataset, we demonstrated the robustness and scalability of our approach, in unveiling sub-populations with a simple and readable interpretation, which exhibit a statistically significant gender-bias. Our analysis further revealed that features such as a users martial status, or capital, may provide partial explanations about the exhibited gender-bias, although they do not account for the discrimination as a whole.

Perspectives, Open Problems and Future Work A logical next step will be to continue the evaluation of our system on various alternative datasets with interesting characteristics, such as multivalued sensitive features or algorithmic outputs, or considerations about the user's utility. We plan on building a multipurpose system, capable of uncovering discriminatory practices in a large variety of use-cases, including utility constraints and potential external knowledge sources. From a technical perspective, our solution should propose a variety of methods for cluster analysis, generation of interpretable hypotheses, and robust statistical testing.

Getting a better understanding of the concept of individual fairness, introduced by Dwork et al. [17], appears to be an important open problem. As we briefly discussed, the current definition, that relies on a generic notion of user-similarity as well as on a notion of closeness in outcome probabilities, still seems to suffer from some important limitations. It is for instance unclear, how user-utility measures would fit into this framework, or whether probabilities of outcomes are the right 'resource', over which equity should be guaranteed.

Finally, an extremely important aspect of the algorithmic fairness question that this work has not touched upon, is that of discrimination prevention. So far, research in this area has focused on building specific types of classifiers, that attain high accuracy while maintaining low bias, when trained on potentially biased data. Most of the proposed techniques rely on fairness measures limited to binary features and algorithmic outcomes, and do not take into account any form of user or business utility. Incorporating utility notions, especially business requirements that justify some algorithmic bias, into discrimination prevention is a critical open problem in this field. We believe that the approaches and measures introduced in this work can provide foundations for the development of mechanisms achieving this goal.

In this context, a question which seems to have received little attention so far is that of choosing appropriate metrics, for quantifying the performance of a 'fair' classifier. While the vast majority of the machine learning literature has focused primarily on optimizing the predictive accuracy of classifiers, the notions of *comprehensibility* and *interpretability* of classification models have recently re-surfaced, most prominently in tasks related to personalized medicine [43] or credit scoring [46], with regard to concerns on *trust*, *accountability* and *justifiability* in algorithmic decision-making [24]. Following our own discussions on the comprehensibility of various discrimination contexts, as well as on the justifiability of genuine business requirements that explain an algorithmic bias, we believe that the interpretability of a classification model is an important aspect to consider, in order to limit the discriminatory potential of algorithmic decision making on a big-data scale.

References

[1] Civil Rights Act of 1964 § 7, 42 U.S.C. §2000e et seq, 1964.

[2] Council of Europe: European Court of Human Rights, Handbook on European non-discrimination law, 2011.

[3] H. Abdi. The Bonferonni and Šidák corrections for multiple comparisons. In N.J. Salkind, editor, *Encyclopedia of measurement and statistics*, Encyclopedia of Measurement and Statistics. SAGE Publications, 2007.

[4] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, pages 247–255, New York, NY, USA, 2001. ACM.

[5] Anthony B Atkinson. On the measurement of inequality. *Journal of economic theory*, 2(3):244–263, 1970.

[6] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Available at SSRN 2477899*, 2014.

[7] Adam Barth, Anupam Datta, John C Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–pp. IEEE, 2006.

[8] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

[9] P. J. Bickel, E. A. Hammel, and J. W. O'Connell. Sex bias in graduate admissions: Data from Berkeley. *Science*, 187(4175):398–404, 1975.

[10] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[11] John Broome. Fairness. *Proceedings of the Aristotelian Society*, 91:pp. 87–101, 1990.

[12] Toon Calders and Sicco Verwer. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

[13] Equal Employment Opportunity Commission. Information on impact (§ 1607.4), Uniform Guidelines on Employee Selection Procedure, 1978.

[14] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *arXiv preprint arXiv:1408.6491*, 2014.

[15] Anupam Datta. Privacy through accountability: A computer science perspective. In Raja Natarajan, editor, *Distributed Computing and Internet Technology*, volume 8337 of *Lecture Notes in Computer Science*, pages 43–49. Springer International Publishing, 2014.

[16] Cynthia Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[17] 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*, ITCS '12, pages 214–226, New York, NY, USA, 2012. ACM.

[18] Deniz Erdogmus and Jose C Principe. Lower and upper bounds for misclassification probability based on renyi's information. *Journal of VLSI signal processing systems for signal, image and video technology*, 37(2-3):305–317, 2004.

[19] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *Neural Networks, IEEE Transactions on*, 20(2):189–201, 2009.

[20] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222. ACM, 2003.

[21] Meir Feder and Neri Merhav. Relations between entropy and error probability. *Information Theory, IEEE Transactions on*, 40(1):259–266, 1994.

[22] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. *ArXiv e-prints*, December 2014.

[23] D. Freedman, R. Pisani, and R. Purves. *Statistics*. International student edition. W.W. Norton & Company, 2007.

[24] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

[25] Joseph L Gastwirth. Statistical reasoning in the legal setting. *The American Statistician*, 46(1):55–69, 1992.

[26] Roxana Geambasu. Private communication.

[27] Tristin K Green. Discrimination in workplace dynamics: Toward a structural account of disparate treatment theory. *Harv. CR-CLL Rev.*, 38:91, 2003.

[28] Susan S Grover. Business necessity defense in disparate impact discrimination cases, the. *Ga. L. Rev.*, 30:387, 1995.

[29] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[30] Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 25(7):1445–1459, 2013.

[31] Moritz Hardt. How big data is unfair. Understanding sources of unfairness in data driven decision making. `https://medium.com/@mrtz/how-big-data-is-unfair-9aa544d739de`, September 2014.

[32] Peter Harremoës and Gábor Tusnády. Information divergence is more χ^2-distributed than the χ^2-statistics. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 533–537. IEEE, 2012.

[33] Jesse Hoey. The two-way likelihood ratio (G) test and comparison to two-way chi squared test. *arXiv preprint arXiv:1206.4881*, 2012.

[34] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[35] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[36] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–6. IEEE, 2009.

[37] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 869–874. IEEE, 2010.

[38] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[39] Rogier A Kievit, Willem E Frankenhuis, Lourens J Waldorp, and Denny Borsboom. Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology*, 4, 2013.

[40] Ivan Kojadinovic. On the use of mutual information in data analysis: an overview. In *Applied Stochastic Models and Data Analysis (ASMDA 2005)*, Brest, France, 2005.

[41] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[42] Mathias Lécuyer, Guillaume Ducoffe, Francis Lan, Andrei Papancea, Theofilos Petsios, Riley Spahn, Augustin Chaintreau, and Roxana Geambasu. XRay: Enhancing the web's transparency with differential correlation. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 49–64, San Diego, CA, 2014. USENIX Association.

[43] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Department of Statistics Technical Report tr608, University of Washington, 2014.

[44] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 502–510. ACM, 2011.

[45] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. From the information bottleneck to the privacy funnel. In *Information Theory Workshop (ITW), 2014 IEEE*, pages 501–505. IEEE, 2014.

[46] David Martens, Jan Vanthienen, Wouter Verbeke, and Bart Baesens. Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793, 2011.

[47] John H McDonald. *Handbook of Biological Statistics*, volume 3. 3rd ed. Sparky House Publishing, Baltimore, Maryland, 2014.

[48] Wes McKinney. Data structures for statistical computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[49] Thomas E Nichols and Andrew P Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1):1–25, 2002.

[50] Ramona Paetzold and Steven L. Willborn. *The Statistics of Discrimination: Using Statistical Evidence in Discrimination Cases.* McGraw-Hill/Shepard's, 1994.

[51] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[53] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.

[54] Jennifer L Peresie. Toward a coherent test for disparate impact discrimination. *Ind. LJ*, 84:773, 2009.

[55] Fernando Perez and Brian E Granger. IPython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, 2007.

[56] Pamela L Perry. Balancing equal imployment opportunities with employers' legitimate discretion: The business necessity response to disparate impact discrimination under title vii. *Indus. Rel. LJ*, 12:1, 1990.

[57] Daniel L Rubinfeld. Econometrics in the courtroom. *Columbia Law Review*, pages 1048–1097, 1985.

[58] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Data mining for discrimination discovery. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(2):9, 2010.

[59] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Dcube: Discrimination discovery in databases. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1127–1130. ACM, 2010.

[60] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Integrating induction and deduction for finding evidence of discrimination. *Artificial Intelligence and Law*, 18(1):1–43, 2010.

[61] Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.

[62] George Sher. What makes a lottery fair? *Noûs*, pages 203–216, 1980.

[63] Elaine W Shoben. Differential pass-fail rates in employment testing: Statistical proof under title vii. *Harvard Law Review*, pages 793–813, 1978.

[64] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

[65] R.R. Sokal and F.J. Rohlf. *Biometry*. W. H. Freeman, 1995.

[66] Andrew C Spiropoulos. Defining the business necessity defense to the disparate impact cause of action: Finding the golden mean. *NCL Rev.*, 74:1479, 1995.

[67] Latanya Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10, 2013.

[68] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *The 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999.

[69] Michael Carl Tschantz, Amit Datta, Anupam Datta, and Jeannette M Wing. A methodology for information flow experiments. *arXiv preprint arXiv:1405.2376*, 2014.

[70] Michael Carl Tschantz, Anupam Datta, and Jeannette M Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 176–190. IEEE, 2012.

[71] Jennifer Valentino-Devries, Jeremy Singer-Vine, and Ashkan Soltani. Websites vary prices, deals based on users' information. *The Wall Street Journal*, December 24, 2012. http://www.wsj.com/articles/SB10001424127887323777204578189391813881534.

[72] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[73] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 325–333, 2013.

[74] Indre Zliobaite, Faisal Kamiran, and Toon Calders. Handling conditional discrimination. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 992–1001. IEEE, 2011.

Appendix

A Discrimination Law Background

To introduce legal approaches for discrimination protection, we focus on the 'European non-discrimination law' [2], in use in European Union, and on Title VII of the American Civil Rights Acts of 1962 [1] and 1991, on discrimination in employment.

Discrimination law in the EU is not defined with a particular context in mind. Its aim is to 'allow all individuals an equal and fair prospect to access opportunities available in a society' [2]. The law introduces *protected grounds*, upon which discrimination should be prohibited, namely 'sex, sexual orientation, disability, age, race, ethnic origin, national origin and religion' [2]. In contrast, Title VII focuses exclusively on discrimination in employment, based on 'race, color, religion, sex and national origin' [1]. Additional acts, broadening the scope of anti-discrimination legislation, include the *Civil Rights Act of 1968* on discrimination in housing, the *Age Discrimination Act* and the *Americans with Disabilities Act of 1990.*

Both the European and American laws similarly distinguish between two theories of discrimination. The EU law defines *Direct Discrimination* and *Indirect Discrimination*, which are roughly analogous to the theories of *Disparate Treatment* and *Disparate Impact* under Title VII.

Direct Discrimination and Disparate Treatment Direct discrimination or disparate treatment refer to the situation where an *individual* is treated unfavorably, compared to other individuals in a similar situation, on the basis of a protected attribute. A key property of direct discrimination and disparate impact is the *intent* to discriminate, implying that there should be a *causal link* between the treatment and the protected attribute. A case of intentional discrimination towards an individual will usually be made by comparing the individual's treatment to that of another individual in a similar situation, where the main difference between the two is a protected attribute.

Indirect Discrimination and Disparate Impact In contrast, indirect discrimination or disparate impact refer to situations, where a facially 'neutral rule, criterion or practice' [2], has a discriminatory effect or impact on members of a protected group, compared to other groups in a similar situation. In such a case, it is thus not necessary to prove an *intent* to discriminate, but rather, provide evidence that the members of a protected group are disproportionately affected by some practice.

The European law advocates the use of 'statistical evidence' for establishing indirect discrimination, but no 'strict threshold requirement' is provided [2]. In the US, the 'Uniform Guidelines on Employee Selection Procedures' introduce the so-called *four-fifths rule* as a guideline for the enforcement of Title VII. The rule states that a ratio in selection rates of less than $\frac{4}{5}$ (or more than $\frac{5}{4}$) between two protected groups will generally be regarded as evidence of disparate impact. However, it is emphasized that this threshold is not absolute, and that the statistical significance of the adverse impact should also be evaluated [13, 6].

Business Defense Both the European and American discrimination laws accept that a differential treatment may be justified, if it pursues a legitimate and necessary business aim.

In the EU, this general defense strategy is limited to *indirect discrimination*, whenever the 'provision, criterion or practice is objectively justified by a legitimate aim, and the means of achieving that aim are appropriate and necessary' [2]. For the particular case of discrimination in employment, discrimination based on a characteristic related to protected features is justified, if the 'characteristic constitutes a genuine and determining occupational requirement' [2]. Similarly, under Title VII, an employer may justify disparate impact, by showing that the discriminatory practice is due to legitimate requirements for a particular job [28]. This is commonly called the *business-necessity defense.*

Interpretation in Previous Work The work of Ruggieri et al. [60, 58] has distinguished between direct and indirect discrimination, depending on whether sensitive features S were part of the collected data. They refer to direct discrimination, when a direct dependency between S and O can be established, and to indirect discrimination, when additional external knowledge about S has to be used to estimate this dependency. We note that this approach fails to take into account the notion of *causality* or *intent*, that is essential in the legal definitions of direct discrimination and disparate treatment. Dwork et al. [17, 73] propose to distinguish between *individual-fairness*, asking that two similar individuals be treated similarly, and *group-fairness*, asking that two protected groups be treated similarly on average. This approach also does not consider the distinction between intentional and unintentional discrimination. A popular method for detecting causal effects between two features is a *randomized experiment.* This is used for instance in [14], to detect direct discrimination in ad-targeting.

Previous works have also considered various interpretations of the notion of 'similar situation', appearing in EU and American laws. Ruggieri et al. [60, 58] use frequent itemset mining to discover *discrimination contexts*, in which a significant algorithmic bias is exhibited. Luong et al. [44] propose to measure discrimination in small subsets of similar users, where similarity is defined using a k-NN clustering. Finally, Dwork et al. [17, 73] propose a definition of individual fairness based upon on an arbitrary user similarity metric.

Interpretation in This Work In this Thesis, we consider algorithmic discrimination as a case of *indirect discrimination.* Indeed, we may fairly assume that the algorithm's design goal is not to actively and intentionally discriminate, but rather that its application may lead to a differential effect on certain protected groups. We thus focus on detecting dependencies between sensitive features and algorithmic outputs, and make no conclusions regarding the causality of these effects. In accordance with legal practices, we make use of statistical tests to discover algorithmic biases, both in terms of significant differences in outcome proportions, as well as in significant absolute differences. Compared to previous works, we also interpret the notion of differential effect in a more broader sense, to include situations where users are treated similarly but perceive different utilities from algorithmic decisions.

In order to evaluate an algorithm's bias over groups of similar users, we first learn a simple and interpretable model of the algorithm's decision-process, and then cluster users based on important non-sensitive features, with respect to the classification task at hand.