Learning Fair Naive Bayes Classifiers by Discovering and Eliminating Discrimination Patterns

YooJung Choi,[1*] **Golnoosh Farnadi,**[2,3*] **Behrouz Babaki,**[4*] **and Guy Van den Broeck**[1]
[1]University of California, Los Angeles, [2]Mila, [3]Université de Montréal, [4]Polytechnique Montréal
yjchoi@cs.ucla.edu, farnadig@mila.quebec, behrouz.babaki@polymtl.ca, guyvdb@cs.ucla.edu

Abstract

As machine learning is increasingly used to make real-world decisions, recent research efforts aim to define and ensure fairness in algorithmic decision making. Existing methods often assume a fixed set of observable features to define individuals, but lack a discussion of certain features not being observed at test time. In this paper, we study fairness of naive Bayes classifiers, which allow partial observations. In particular, we introduce the notion of a discrimination pattern, which refers to an individual receiving different classifications depending on whether some sensitive attributes were observed. Then a model is considered fair if it has no such pattern. We propose an algorithm to discover and mine for discrimination patterns in a naive Bayes classifier, and show how to learn maximum-likelihood parameters subject to these fairness constraints. Our approach iteratively discovers and eliminates discrimination patterns until a fair model is learned. An empirical evaluation on three real-world datasets demonstrates that we can remove exponentially many discrimination patterns by only adding a small fraction of them as constraints.

1 Introduction

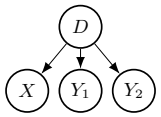
With the increasing societal impact of machine learning come increasing concerns about the fairness properties of machine learning models and how they affect decision making. For example, concerns about fairness come up in policing (Mohler et al. 2018), recidivism prediction (Chouldechova 2017), insurance pricing (Kusner et al. 2017), hiring (Datta, Tschantz, and Datta 2015), and credit rating (Henderson et al. 2015). The algorithmic fairness literature has proposed various solutions, from limiting the disparate treatment of similar individuals to giving statistical guarantees on how classifiers behave towards different populations. Key approaches include individual fairness (Dwork et al. 2012; Zemel et al. 2013), statistical parity, disparate impact and group fairness (Kamishima et al. 2012; Feldman et al. 2015; Chouldechova 2017), counterfactual fairness (Kusner et al. 2017), preference-based fairness (Zafar et al. 2017a), relational fairness (Farnadi, Babaki, and Getoor 2018), and equality of opportunity (Hardt et al. 2016). The goal in these works is usually to assure the fair treatment of individuals or groups that are identified by sensitive attributes.

In this paper, we study fairness properties of probabilistic classifiers that represent joint distributions over the features and decision variable. In particular, Bayesian network classifiers treat the classification or decision-making task as a probabilistic inference problem: given observed features, compute the probability of the decision variable. Such models have a unique ability that they can naturally handle missing features, by simply marginalizing them out of the distribution when they are not observed at prediction time. Hence, a Bayesian network classifier effectively embeds exponentially many classifiers, one for each subset of observable features. We ask whether such classifiers exhibit patterns of discrimination where similar individuals receive markedly different outcomes purely because they disclosed a sensitive attribute.

The first key contribution of this paper is an algorithm to verify whether a Bayesian classifier is fair, or else to mine the classifier for discrimination patterns. We propose two alternative criteria for identifying the most important discrimination patterns that are present in the classifier. We specialize our pattern miner to efficiently discover discrimination patterns in naive Bayes models using branch-and-bound search. These classifiers are often used in practice because of their simplicity and tractability, and they allow for the development of effective bounds. Our empirical evaluation shows that naive Bayes models indeed exhibit vast numbers of discrimination patterns, and that our pattern mining algorithm is able to find them by traversing only a small fraction of the search space.

The second key contribution of this paper is a parameter learning algorithm for naive Bayes classifiers that ensures that no discrimination patterns exist in the the learned distribution. We propose a signomial programming approach to eliminate individual patterns of discrimination during maximum-likelihood learning. Moreover, to efficiently eliminate the exponential number of patterns that could exist in a naive Bayes classifier, we propose a cutting-plane approach that uses our discrimination pattern miner to find and iteratively eliminate discrimination patterns until the entire learned model is fair. Our empirical evaluation shows that this process converges in a small number of iteration, effectively removing millions of discrimination patterns. Moreover, the learned fair mod-

[*]Equal contribution



$P(d)$
0.2

D	$P(x\|D)$
+	0.8
−	0.5

D	$P(y_1\|D)$
+	0.7
−	0.1

D	$P(y_2\|D)$
+	0.8
−	0.3

Figure 1: Naive Bayes classifier with a sensitive attribute X and non-sensitive attributes Y_1, Y_2

els are of high quality, achieving likelihoods that are close to the best likelihoods attained by models with no fairness constraints. Our method also achieves higher accuracy than other methods of learning fair naive Bayes models.

2 Problem Formalization

We use uppercase letters for random variables and lowercase letters for their assignments. Sets of variables and their joint assignments are written in bold. Negation of a binary assignment x is denoted $\bar{x}$, and $\mathbf{x} \models \mathbf{y}$ means that $\mathbf{x}$ logically implies $\mathbf{y}$. Concatenation of sets $\mathbf{XY}$ denotes their union.

Each individual is characterized by an assignment to a set of discrete variables $\mathbf{Z}$, called attributes or features. Assignment d to a binary decision variable D represents a decision made in favor of the individual (e.g., a loan approval). A set of *sensitive attributes* $\mathbf{S} \subset \mathbf{Z}$ specifies a group of entities protected often by law, such as gender and race. We now define the notion of a discrimination pattern.

Definition 1. *Let P be a distribution over $D \cup \mathbf{Z}$. Let $\mathbf{x}$ and $\mathbf{y}$ be joint assignments to $\mathbf{X} \subseteq \mathbf{S}$ and $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$, respectively. The* degree of discrimination *of $\mathbf{xy}$ is:*

$$\Delta_{P,d}(\mathbf{x}, \mathbf{y}) \triangleq P(d \,|\, \mathbf{xy}) - P(d \,|\, \mathbf{y}).$$

The assignment $\mathbf{y}$ identifies a group of similar individuals, and the degree of discrimination quantifies how disclosing sensitive information $\mathbf{x}$ affects the decision for this group. Note that sensitive attributes missing from $\mathbf{x}$ can still appear in $\mathbf{y}$. We drop the subscripts P, d when clear from context.

Definition 2. *Let P be a distribution over $D \cup \mathbf{Z}$, and $\delta \in [0, 1]$ a threshold. Joint assignments $\mathbf{x}$ and $\mathbf{y}$ form a* discrimination pattern *w.r.t. P and δ if: (1) $\mathbf{X} \subseteq \mathbf{S}$ and $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$; and (2) $|\Delta_{P,d}(\mathbf{x}, \mathbf{y})| > \delta$.*

Intuitively, we do not want information about the sensitive attributes to significantly affect the probability of getting a favorable decision. Let us consider two special cases of discrimination patterns. First, if $\mathbf{Y} = \emptyset$, then a small discrimination score $|\Delta(\mathbf{x}, \emptyset)|$ can be interpreted as an approximation of statistical parity, which is achieved when $P(d \,|\, \mathbf{x}) = P(d)$. For example, the naive Bayes network in Figure 1 satisfies approximate parity for $\delta = 0.2$ as $|\Delta(x, \emptyset)| = 0.086 \leq \delta$ and $|\Delta(\bar{x}, \emptyset)| = 0.109 \leq \delta$. Second, suppose $\mathbf{X} = \mathbf{S}$ and $\mathbf{Y} = \mathbf{Z} \setminus \mathbf{S}$. Then bounding $|\Delta(\mathbf{x}, \mathbf{y})|$ for all joint states $\mathbf{x}$ and $\mathbf{y}$ is equivalent to enforcing individual fairness assuming two individuals are considered similar if their non-sensitive attributes $\mathbf{y}$ are equal. The network in Figure 1 is also individually fair for $\delta = 0.2$ because $\max_{xy_1y_2} |\Delta(x, y_1y_2)| = 0.167 \leq \delta$.[1] We discuss these connections more in Section 5.

Algorithm 1 DISC-PATTERNS$(\mathbf{x}, \mathbf{y}, \mathbf{E})$

Input: P : Distribution over $D \cup \mathbf{Z}$, δ : discrimination threshold
Output: Discrimination patterns L
Data: $\mathbf{x} \leftarrow \emptyset, \mathbf{y} \leftarrow \emptyset, \mathbf{E} \leftarrow \emptyset, L \leftarrow []$

1: **for** all assignments z to some selected variable $Z \in \mathbf{Z} \setminus \mathbf{XYE}$ **do**
2: **if** $Z \in \mathbf{S}$ **then**
3: **if** $|\Delta(\mathbf{x}z, \mathbf{y})| > \delta$ **then** add $(\mathbf{x}z, \mathbf{y})$ to L
4: **if** UB$(\mathbf{x}z, \mathbf{y}, \mathbf{E}) > \delta$ **then** DISC-PATTERNS$(\mathbf{x}z, \mathbf{y}, \mathbf{E})$
5: **if** $|\Delta(\mathbf{x}, \mathbf{y}z)| > \delta$ **then** add $(\mathbf{x}, \mathbf{y}z)$ to L
6: **if** UB$(\mathbf{x}, \mathbf{y}z, \mathbf{E}) > \delta$ **then** DISC-PATTERNS$(\mathbf{x}, \mathbf{y}z, \mathbf{E})$
7: **if** UB$(\mathbf{x}, \mathbf{y}, \mathbf{E} \cup \{Z\}) > \delta$ **then** DISC-PATTERNS$(\mathbf{x}, \mathbf{y}, \mathbf{E} \cup \{Z\})$

Even though the example network is considered (approximately) fair at the group level nor at the individual level with fully observed features, it may still produce a discrimination pattern. In particular, $|\Delta(\bar{x}, y_1)| = 0.225 > \delta$. That is, a person with $\bar{x}$ and y_1 observed and the value of Y_2 undisclosed would receive a much more favorable decision had they not disclosed X as well. Hence, naturally we wish to ensure that there exists no discrimination pattern across all subsets of observable features.

Definition 3. *A distribution P is* δ-fair *if there exists no discrimination pattern w.r.t P and δ.*

Although our notion of fairness applies to any distribution, finding discrimination patterns can be computationally challenging: computing the degree of discrimination involves probabilistic inference, which is hard in general, and a given distribution may have exponentially many patterns. In this paper, we demonstrate how to discover and eliminate discrimination patterns of a naive Bayes classifier effectively by exploiting its independence assumptions. Concretely, we answer the following questions: (1) Can we certify that a classifier is δ-fair?; (2) If not, can we find the most important discrimination patterns?; (3) Can we learn a naive Bayes classifier that is entirely δ-fair?

3 Discovering Discrimination Patterns and Verifying δ-fairness

This section describes our approach to finding discrimination patterns or checking that there are none.

3.1 Searching for Discrimination Patterns

One may naively enumerate all possible patterns and compute their degrees of discrimination. However, this would be very inefficient as there are exponentially many subsets and assignments to consider. We instead use branch-and-bound search to more efficiently decide if a model is fair.

Algorithm 1 finds discrimination patterns. It recursively adds variable instantiations and checks the discrimination

[1] The highest discrimination score is observed at $\bar{x}$ and $y_1\bar{y}_2$, with $\Delta(\bar{x}, y_1\bar{y}_2) = -0.167$.

score at each step. If the input distribution is δ-fair, the algorithm returns no pattern; otherwise, it returns the set of all discriminating patterns. Note that computing Δ requires probabilistic inference on distribution P. This can be done efficiently for large classes of graphical models (Darwiche 2009; Poon and Domingos 2011; Dechter 2013; Rahman, Kothalkar, and Gogate 2014; Kisa et al. 2014), and particularly for naive Bayes networks, which will be our main focus.

Furthermore, the algorithm relies on a good upper bound to prune the search tree and avoid enumerating all possible patterns. Here, $\mathrm{UB}(\mathbf{x}, \mathbf{y}, \mathbf{E})$ bounds the degree of discrimination achievable by observing more features after $\mathbf{xy}$ while excluding features $\mathbf{E}$.

Proposition 1. *Let P be a naive Bayes distribution over $D \cup \mathbf{Z}$, and let $\mathbf{x}$ and $\mathbf{y}$ be joint assignments to $\mathbf{X} \subseteq \mathbf{S}$ and $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$. Let $\mathbf{x}'_u$ (resp. $\mathbf{x}'_l$) be an assignment to $\mathbf{X}' = \mathbf{S} \setminus \mathbf{X}$ that maximizes (resp. minimizes) $P(d \mid \mathbf{xx}')$. Suppose $l, u \in [0,1]$ such that $l \leq P(d \mid \mathbf{yy}') \leq u$ for all possible assignments $\mathbf{y}'$ to $\mathbf{Y}' = \mathbf{Z} \setminus (\mathbf{XY})$. Then the degrees of discrimination for all patterns $\mathbf{xx}'\mathbf{yy}'$ that extend $\mathbf{xy}$ are bounded as follows:*

$$\min_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_l \mid d), P(\mathbf{x}\mathbf{x}'_l \mid \overline{d}), \gamma\right) \leq \Delta_{P,d}(\mathbf{xx}', \mathbf{yy}') \leq \max_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_u \mid d), P(\mathbf{x}\mathbf{x}'_u \mid \overline{d}), \gamma\right),$$

where $\widetilde{\Delta}(\alpha, \beta, \gamma) \triangleq \frac{\alpha\gamma}{\alpha\gamma + \beta(1-\gamma)} - \gamma$.

Here, $\widetilde{\Delta} : [0,1]^3 \to [0,1]$ is introduced to relax the discrete problem of minimizing or maximizing the degree of discrimination into a continuous one. In particular, $\widetilde{\Delta}\left(P(\mathbf{x}|d), P(\mathbf{x}|\overline{d}), P(d|\mathbf{y})\right)$ equals the degree of discrimination $\Delta(\mathbf{x}, \mathbf{y})$. This relaxation allows us to compute bounds efficiently, as closed-form solutions. We refer to the Appendix for full proofs and details.

To apply above proposition, we need to find $\mathbf{x}'_u, \mathbf{x}'_l, l, u$ by maximizing/minimizing $P(d|\mathbf{xx}')$ and $P(d|\mathbf{yy}')$ for a given pattern $\mathbf{xy}$. Fortunately, this can be done efficiently for naive Bayes classifiers.

Lemma 1. *Given a naive Bayes distribution P over $D \cup \mathbf{Z}$, a subset $\mathbf{V} = \{V_i\}_{i=1}^n \subset \mathbf{Z}$, and an assignment $\mathbf{w}$ to $\mathbf{W} \subseteq \mathbf{Z} \setminus \mathbf{V}$, we have:* $\arg\max_{\mathbf{v}} P(d|\mathbf{vw}) = \left\{\arg\max_{v_i} P(v_i|d)/P(v_i|\overline{d})\right\}_{i=1}^n$.

That is, the joint observation $\mathbf{v}$ that will maximize the probability of the decision can be found by optimizing each variable V_i independently; the same holds when minimizing. Hence, we can use Proposition 1 to compute upper bounds on discrimination scores of extended patterns in linear time.

3.2 Searching for Top-k Ranked Patterns

If a distribution is significantly unfair, Algorithm 1 may return exponentially many discrimination patterns. This is not only very expensive but makes it difficult to interpret the discrimination patterns. Instead, we would like to return a smaller set of "interesting" discrimination patterns.

An obvious choice is to return a small number of discrimination patterns with the highest absolute degree of discrimination. Searching for the k most discriminating patterns can be done with a small modification to Algorithm 1. First, the size of list L is limited to k. The conditions in Lines 3–7 are modified to check the current discrimination score and upper bounds against the smallest discrimination score of patterns in L, instead of the threshold δ.

Nevertheless, ranking patterns by their discrimination score may return patterns of very low probability. For example, the most discriminating pattern of a naive Bayes classifier learned on the COMPAS dataset[2] has a high discrimination score of 0.42, but only has a 0.02% probability of occurring.[3] The probability of a discrimination pattern denotes the proportion of the population (according to the distribution) that could be affected unfairly, and thus a pattern with extremely low probability could be of lesser interest. To address this concern, we propose a more sophisticated ranking of the discrimination patterns that also takes into account the probabilities of patterns.

Definition 4. *Let P be a distribution over $D \cup \mathbf{Z}$. Let $\mathbf{x}$ and $\mathbf{y}$ be joint instantiations to subsets $\mathbf{X} \subseteq \mathbf{S}$ and $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$, respectively. The* divergence score *of $\mathbf{xy}$ is:*

$$\mathrm{Div}_{P,d,\delta}(\mathbf{x}, \mathbf{y}) \triangleq \min_{Q} \mathrm{KL}\left(P \parallel Q\right) \quad \text{s.t. } |\Delta_{Q,d}(\mathbf{x}, \mathbf{y})| \leq \delta,\ P(d\mathbf{z}) = Q(d\mathbf{z}),\ \forall\, d\mathbf{z} \not\models \mathbf{xy} \tag{1}$$

where $\mathrm{KL}\left(P \parallel Q\right) = \sum_{d,\mathbf{z}} P(d\mathbf{z}) \log(P(d\mathbf{z})/Q(d\mathbf{z}))$.

The divergence score assigns to a pattern $\mathbf{xy}$ the minimum Kullback-Leibler (KL) divergence between current distribution P and a hypothetical distribution Q that is fair on the pattern $\mathbf{xy}$ and differs from P only on the assignments that satisfy the pattern (namely $d\mathbf{xy}$ and $\overline{d}\mathbf{xy}$). Informally, the divergence score approximates how much the current distribution P needs to be changed in order for $\mathbf{xy}$ to no longer be a discrimination pattern. Hence, patterns with higher divergence score will tend to have not only higher discrimination score but also higher probabilities.

For instance, the pattern with the highest divergence score [4] on the COMPAS dataset has a discrimination score of 0.19 which is not insignificant, but also has a relatively high probability of 3.33% – more than two orders of magnitude larger than that of the most discriminating pattern. Therefore, such a general pattern could be more interesting for the user studying this classifier.

To find the top-k patterns with the divergence score, we need to be able to compute the score and its upper bound efficiently. The key insights are that KLD is convex and that Q, in Equation 1, can freely differ from P only on one probability value (either that of $d\mathbf{xy}$ or $\overline{d}\mathbf{xy}$). Then:

$$\mathrm{Div}_{P,d,\delta}(\mathbf{x}, \mathbf{y}) = P(d\mathbf{xy}) \log\left(\frac{P(d\mathbf{xy})}{P(d\mathbf{xy}) + r}\right) + P(\overline{d}\mathbf{xy}) \log\left(\frac{P(\overline{d}\mathbf{xy})}{P(\overline{d}\mathbf{xy}) - r}\right), \tag{2}$$

[2] https://github.com/propublica/compas-analysis

[3] The corresponding pattern is $\mathbf{x} = \{\text{White, Married, Female, } > 30 \text{ y/o}\}$, $\mathbf{y} = \{\text{Probation, Pretrial}\}$.

[4] $\mathbf{x} = \{\text{Married, } > 30 \text{ y/o}\}$, $\mathbf{y} = \{\}$.

where $r = 0$ if $|\Delta_{P,d}(\mathbf{x},\mathbf{y})| \leq \delta$; $r = \frac{\delta - \Delta_{P,d}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}$ if $\Delta_{P,d}(\mathbf{x},\mathbf{y}) > \delta$; and $r = \frac{-\delta - \Delta_{P,d}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}$ if $\Delta_{P,d}(\mathbf{x},\mathbf{y}) < -\delta$. Intuitively, r represents the minimum necessary change to $P(d\mathbf{xy})$ for $\mathbf{xy}$ to be non-discriminating in the new distribution. Note that the smallest divergence score of 0 is attained when the pattern is already fair.

Lastly, we refer to the Appendix for two upper bounds of the divergence score, which utilize the bound on discrimination score of Proposition 1 and can be computed efficiently using Lemma 1.

3.3 Empirical Evaluation of Discrimination Pattern Miner

In this section, we report the experimental results on the performance of our pattern mining algorithms. All experiments were run on an AMD Opteron 275 processor (2.2GHz) and 4GB of RAM running Linux Centos 7. Execution time is limited to 1800 seconds.

Data and pre-processing. We use three datasets: The *Adult* dataset and *German* dataset are used for predicting income level and credit risk, respectively, and are obtained from the UCI machine learning repository[5]; the *COMPAS* dataset is used for predicting recidivism. These datasets have been commonly studied regarding fairness and were shown to exhibit some form of discrimination by several previous works (Luong, Ruggieri, and Turini 2011; Larson et al. 2016; Tramer et al. 2017; Salimi et al. 2019). As pre-processing, we removed unique features (e.g. names of individuals) and duplicate features.[6] See Table 1 for a summary.

Q1. Does our pattern miner find discrimination patterns more efficiently than by enumerating all possible patterns? We answer this question by inspecting the fraction of all possible patterns that our pattern miner visits during the search. Table 1 shows the results on three datasets, using two rank heuristics (discrimination and divergence) and three threshold values (0.01, 0.05, and 0.1). The results are reported for mining the top-k patterns when k is 1, 10, and 100. A naive method has to enumerate all possible patterns to discover the discriminating ones, while our algorithm visits only a small fraction of patterns (e.g., one in every several millions on the German dataset).

Q2. Does the divergence score find discrimination patterns with both a high discrimination score and high probability? Figure 2 shows the *probability* and *discrimination score* of all patterns in the COMPAS dataset. The top-10 patterns according to three measures (discrimination score, divergence score, and probability) are highlighted in the figure. The observed trade-off between probability and discrimination score indicates that picking the top patterns according to each measure will yield low quality patterns according to the other measure. The divergence score, however, balances the two measures and returns patterns that have high probability and discrimination scores. Also observe that the patterns selected by the divergence score lie in the Pareto front. This in fact always holds by the definition of this heuristic; fixing the probability and increasing the discrimination score will also increase the divergence score, and vice versa.

[5]https://archive.ics.uci.edu/ml

[6]The processed data, code, and Appendix are available at https://github.com/UCLA-StarAI/LearnFairNB.

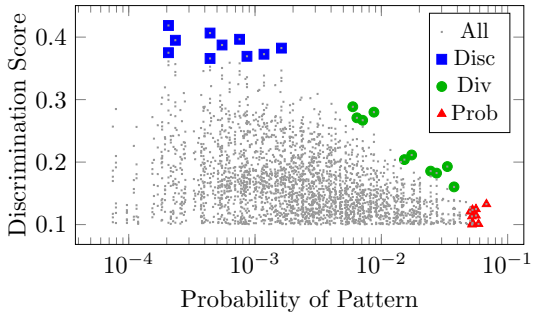


Figure 2: Discrimination patterns with $\delta = 0.1$ for the max-likelihood NB classifier on COMPAS.

4 Learning Fair Naive Bayes Classifiers

We now describe our approach to learning the maximum-likelihood parameters of a naive Bayes model from data while eliminating discrimination patterns. A common approach to learning naive Bayes models with certain properties is to formulate it as an optimization problem of certain form, for which efficient solvers are available (Khosravi et al. 2019). We formulate the learning subject to fairness constraints as a signomial program, which has the following form:

$$\text{minimize } f_0(x), \quad \text{s.t.} \quad f_i(x) \leq 1, \quad g_j(x) = 1 \quad \forall\, i,j$$

where each f_i is signomial while g_j is monomial. A *signomial* is a function of the form $\sum_k c_k x_1^{a_{1k}} \cdots x_n^{a_{1n}}$ defined over real positive variables $x_1 \dots x_n$ where $c_k, a_{ij} \in \mathbb{R}$; a *monomial* is of the form $c x_1^{a_1} \cdots x_n^{a_n}$ where $c > 0$ and $a_i \in \mathbb{R}$. Signomial programs are not globally convex, but a locally optimal solution can be computed efficiently, unlike the closely related class of geometric programs, for which the globally optimum can be found efficiently (Ecker 1980).

4.1 Parameter Learning with Fairness Constraints

The likelihood of a Bayesian network given data $\mathcal{D}$ is $P_\theta(\mathcal{D}) = \prod_i \theta_i^{n_i}$ where n_i is the number of examples in $\mathcal{D}$ that satisfy the assignment corresponding to parameter θ_i. To learn the maximum-likelihood parameters, we minimize the inverse of likelihood which is a monomial: $\theta_{\text{ml}} = \arg\min_\theta \prod_i \theta_i^{-n_i}$. The parameters of a naive Bayes network with binary class consist of $\theta_d, \theta_{\bar{d}}$, and $\theta_{z\,|\,d}, \theta_{z\,|\,\bar{d}}$ for all z.

Next, we show the constraints for our optimization problem. To learn a valid distribution, we need to ensure that probabilities are non-negative and sum to one. The former assumption is inherent to signomial programs. To enforce the latter, for each instantiation d and feature Z, we need that $\sum_z \theta_{z\,|\,d} = 1$, or as signomial inequality constraints: $\sum_z \theta_{z\,|\,d} \leq 1$ and $2 - \sum_z \theta_{z\,|\,d} \leq 1$.

Finally, we derive the constraints to ensure that a given pattern $\mathbf{xy}$ is non-discriminating.

Dataset Statistics						Proportion of search space explored					
						Divergence			Discrimination		
Dataset	Size	S	N	# Pat.	k	$\delta = 0.01$	$\delta = 0.05$	$\delta = 0.10$	$\delta = 0.01$	$\delta = 0.05$	$\delta = 0.10$
COMPAS	48,834	4	3	15K	1	6.387e-01	5.634e-01	3.874e-01	8.188e-03	8.188e-03	8.188e-03
					10	7.139e-01	5.996e-01	4.200e-01	3.464e-02	3.464e-02	3.464e-02
					100	8.222e-01	6.605e-01	4.335e-01	9.914e-02	9.914e-02	9.914e-02
Adult	32,561	4	9	11M	1	3.052e-06	7.260e-06	1.248e-05	2.451e-04	2.451e-04	2.451e-04
					10	7.030e-06	1.154e-05	1.809e-05	2.467e-04	2.467e-04	2.467e-04
					100	1.458e-05	1.969e-05	2.509e-05	2.600e-04	2.600e-04	2.597e-04
German	1,000	4	16	23B	1	5.075e-07	2.731e-06	2.374e-06	7.450e-08	7.450e-08	7.450e-08
					10	9.312e-07	3.398e-06	2.753e-06	1.592e-06	1.592e-06	1.592e-06
					100	1.454e-06	4.495e-06	3.407e-06	5.897e-06	5.897e-06	5.897e-06

Table 1: Data statistics (number of training instances, sensitive features S, non-sensitive features N, and potential patterns) and the proportion of patterns explored during the search, using the *Divergence* and *Discrimination* scores as rankings.

Proposition 2. *Let P_θ be a naive Bayes distribution over $D \cup \mathbf{Z}$, and let $\mathbf{x}$ and $\mathbf{y}$ be joint assignments to $\mathbf{X} \subseteq \mathbf{S}$ and $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$. Then $|\Delta_{P_\theta,d}(\mathbf{x},\mathbf{y})| \leq \delta$ for a threshold $\delta \in [0,1]$ iff the following holds:*

$$r_{\mathbf{x}} = \frac{\prod_x \theta_{x \mid \bar{d}}}{\prod_x \theta_{x \mid d}}, \qquad r_{\mathbf{y}} = \frac{\theta_{\bar{d}} \prod_y \theta_{y \mid \bar{d}}}{\theta_d \prod_y \theta_{y \mid d}},$$
$$\left(\frac{1-\delta}{\delta}\right) r_{\mathbf{x}} r_{\mathbf{y}} - \left(\frac{1+\delta}{\delta}\right) r_{\mathbf{y}} - r_{\mathbf{x}} r_{\mathbf{y}}^2 \leq 1,$$
$$-\left(\frac{1+\delta}{\delta}\right) r_{\mathbf{x}} r_{\mathbf{y}} + \left(\frac{1-\delta}{\delta}\right) r_{\mathbf{y}} - r_{\mathbf{x}} r_{\mathbf{y}}^2 \leq 1.$$

Note that above equalities and inequalities are valid signomial program constraints. Thus, we can learn the maximum-likelihood parameters of a naive Bayes network while ensuring a certain pattern is fair by solving a signomial program. Furthermore, we can eliminate multiple patterns by adding the constraints in Proposition 2 for each of them. However, learning a model that is entirely fair with this approach will introduce an exponential number of constraints. Not only does this make the optimization more challenging, but listing all patterns may simply be infeasible.

4.2 Learning δ-fair Parameters

To address the aforementioned challenge of removing an exponential number of discrimination patterns, we propose an approach based on the *cutting plane* method. That is, we iterate between *parameter learning* and *constraint extraction*, gradually adding fairness constraints to the optimization. The parameter learning component is as described in the previous section, where we add the constraints of Proposition 2 for each discrimination pattern that has been extracted so far. For constraint extraction we use the top-k pattern miner presented in Section 3.2. At each iteration, we learn the maximum-likelihood parameters subject to fairness constraints, and find k more patterns using the updated parameters to add to the set of constraints in the next iteration. This process is repeated until the pattern miner finds no more discrimination pattern.

In the worst case, our algorithm may add exponentially many fairness constraints whilst solving multiple optimization problems. However, as we will later show empirically, we can learn a δ-fair model by explicitly enforcing only a small fraction of fairness constraints. The efficacy of our approach depends on strategically extracting patterns that are significant in the overall distribution. Here, we again use a ranking by discrimination or divergence score, which we also evaluate empirically.

4.3 Empirical Evaluation of δ-fair Learner

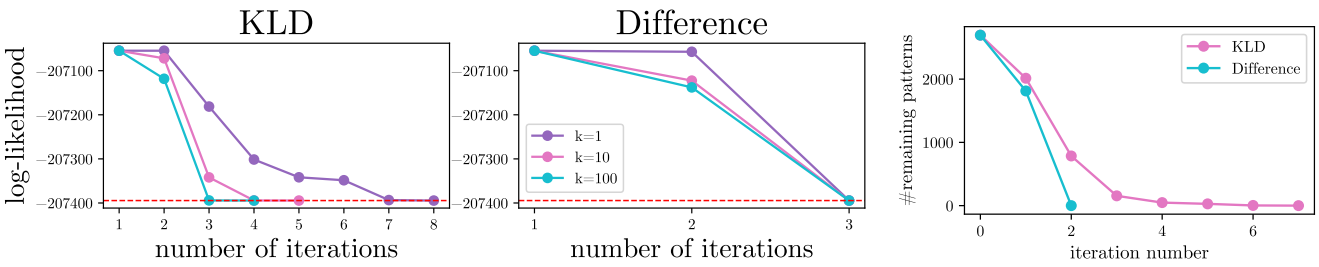
We will now evaluate our iterative algorithm for learning δ-fair naive Bayes models. We use the same datasets and hardware as in Section 3.3. To solve the signomial programs, we use *GPkit*, which finds local solutions to these problems using a convex optimization solver as its backend.[7] Throughout our experiments, Laplace smoothing was used to avoid learning zero probabilities.

Q1. Can we learn a δ-fair model in a small number of iterations while only asserting a small number of fairness constraints? We train a naive Bayes model on the COMPAS dataset subject to δ-fairness constraints. Fig. 3a shows how the iterative method converges to a δ-fair model, whose likelihood is indicated by the dotted line. Our approach converges to a fair model in a few iterations, including only a small fraction of the fairness constraints. In particular, adding only the most discriminating pattern as a constraint at each iteration learns an entirely δ-fair model with only three fairness constraints.[8] Moreover, Fig. 3b shows the number of remaining discrimination patterns after each iteration of learning with $k = 1$. Note that enforcing a single fairness constraint can eliminate a large number of remaining ones. Eventually, a few constraints subsume all discrimination patterns.

We also evaluated our δ-fair learner on the other two datasets; see Appendix for plots. We observed that more than a million discrimination patterns that exist in the unconstrained maximum-likelihood models were eliminated using a few dozen to, even in the worst case, a few thousand fairness constraints. Furthermore, stricter fairness requirements (smaller δ) tend to require more iterations, as would be expected. An interesting observation is that neither of the rankings consistently dominate the other in terms of the number of iterations to converge.

[7] We use Mosek (www.mosek.com) as backend.

[8] There are 2695 discrimination patterns w.r.t. unconstrained naive Bayes on COMPAS and $\delta = 0.1$.



(a) Log-likelihood

(b) Number of remaining patterns

Figure 3: Log-likelihood and the number of remaining discrimination patterns after each iteration of learning on COMPAS dataset with $\delta = 0.1$.

Dataset	Unconstrained	δ-fair	Independent
COMPAS	-207,055	-207,395	-208,639
Adult	-226,375	-228,763	-232,180
German	-12,630	-12,635	-12,649

Table 2: Log-likelihood of models learned without fairness constraints, with the δ-fair learner ($\delta = 0.1$), and by making sensitive variables independent from the decision variable.

Q2. How does the quality of naive Bayes models from our fair learner compare to ones that make the sensitive attributes independent of the decision? and to the best model without fairness constraints? A simple method to guarantee that a naive Bayes model is δ-fair is to make all sensitive variables independent from the target value. An obvious downside is the negative effect on the predictive power of the model. We compare the models learned by our approach with: (1) a maximum-likelihood model with no fairness constraints (unconstrained) and (2) a model in which the sensitive variables are independent of the decision variable, and the remaining parameters are learned using the max-likelihood criterion (independent). These models lie at two opposite ends of the spectrum of the trade-off between fairness and accuracy. The δ-fair model falls between these extremes, balancing approximate fairness and prediction power.

We compare the log-likelihood of these models, shown in Table 2, as it captures the overall quality of a probabilistic classifier which can make predictions with partial observations. The δ-fair models achieve likelihoods that are much closer to those of the unconstrained models than the independent ones. This shows that it is possible to enforce the fairness constraints without a major reduction in model quality.

Dataset	λ =0.5	λ =0.9	λ =0.95	λ =0.99	λ =1.0
COMPAS	2,504	2,471	2,470	3,069	0
Adult	>1e6	661	652	605	0
German	>1e6	3	2	0	0

Table 3: Number of remaining patterns with $\delta = 0.1$ in naive Bayes models trained on discrimination-free data, where λ determines the trade-off between fairness and accuracy in the data repair step (Feldman et al. 2015).

Q3. Do discrimination patterns still occur when learning naive Bayes models from fair data? We first use the data repair algorithm proposed by Feldman et al. (2015) to remove discrimination from data, and learn a naive Bayes model from the repaired data. Table 3 shows the number of remaining discrimination patterns in such model. The results indicate that as long as preserving some degree of accuracy is in the objective, this method leaves lots of discrimination patterns, whereas our method removes all patterns.

dataset	Unconstrained	2NB	Repaired	δ-fair
COMPAS	0.880	0.875	0.878	0.879
Adult	0.811	0.759	0.325	0.827
German	0.690	0.679	0.688	0.696

Table 4: Comparing accuracy of our δ-fair models with two-naive-Bayes method and a naive Bayes model trained on repaired, discrimination-free data.

Q4. How does the performance of δ-fair naive Bayes classifier compare to existing work?

Table 4 reports the 10-fold CV accuracy of our method (δ-fair) compared to a max-likelihood naive Bayes model (unconstrained) and two other methods of learning fair classifiers: the two-naive-Bayes method (2NB) (Calders and Verwer 2010), and a naive Bayes model trained on discrimination-free data using the repair algorithm of Feldman et al. (2015) with $\lambda = 1$. Even though the notion of discrimination patterns was proposed for settings in which predictions are made with missing values, our method still outperforms other fair models in terms of accuracy, a measure better suited for predictions using fully-observed features. Moreover, our method also enforces a stronger definition of fairness than the two-naive-Bayes method which aims to achieve statistical parity, which is subsumed by the notion of discrimination patterns. It is also interesting to observe that our δ-fair NB models perform even better than unconstrained NB models for the Adult and German dataset. Hence, removing discrimination patterns does not necessarily impose an extra cost on the prediction task.

5 Related Work

Most prominent definitions of fairness in machine learning can be largely categorized into *individual fairness* and *group fairness*. Individual fairness is based on the intuition that similar individuals should be treated similarly. For instance, the Lipschitz condition (Dwork et al. 2012) requires that the statistical distance between classifier outputs of two individuals are bounded by a task-specific distance between them. As hinted to in Section 2, our proposed notion of δ-fairness satisfies the Lipschitz condition if two individuals who differ only in the sensitive attributes are considered similar, thus bounding the difference between their outputs by δ. However, our definition cannot represent more nuanced similarity metrics that consider relationships between feature values.

Group fairness aims at achieving equality among populations differentiated by their sensitive attributes. An example of group fairness definition is statistical (demographic) parity, which states that a model is fair if the probability of getting a positive decision is equal between two groups defined by the sensitive attribute, i.e. $P(d|s) = P(d|\bar{s})$ where d and S are positive decision and sensitive variable, respectively. Approximate measures of statistical parity include CV-discrimination score (Calders and Verwer 2010): $P(d|s) - P(d|\bar{s})$; and disparate impact (or p%-rule) (Feldman et al. 2015; Zafar et al. 2017b): $P(d|\bar{s})/P(d|s)$. Our definition of δ-fairness is strictly stronger than requiring a small CV-discrimination score, as a violation of (approximate) statistical parity corresponds to a discrimination pattern with only the sensitive attribute (i.e. empty $\mathbf{y}$). Even though the p%-rule was not explicitly discussed in this paper, our notion of discrimination pattern can be extended to require a small relative (instead of absolute) difference for partial feature observations (see Appendix for details). However, as a discrimination pattern conceptually represents an unfair treatment of an individual based on observing some sensitive attributes, using relative difference should be motivated by an application where the level of unfairness depends on the individual's classification score.

Moreover, statistical parity is inadequate in detecting bias for subgroups or individuals. We resolve such issue by eliminating discrimination patterns for all subgroups that can be expressed as assignments to subsets of features. In fact, we satisfy approximate statistical parity for any subgroup defined over the set of sensitive attributes, as any subgroup can be expressed as a union of joint assignments to the sensitive features, each of which has a bounded discrimination score. Kearns et al. (2018) showed that auditing fairness at this arbitrary subgroup level (i.e. detecting *fairness gerrymandering*) is computationally hard.

Other notions of group fairness include equalized true positive rates (equality of opportunity), false positive rates, or both (equalized odds (Hardt et al. 2016)) among groups defined by the sensitive attributes. These definitions are "oblivious" to features other than the sensitive attribute, and focus on equalizing measures of classifier performance assuming all features are always observed. On the other hand, our method aims to ensure fairness when classifications may be made with missing features. Moreover, our method still applies in decision making scenarios where a true label is not well defined or hard to observe.

Our approach differs from causal approaches to fairness (Kilbertus et al. 2017; Kusner et al. 2017; Russell et al. 2017) which are more concerned with the causal mechanism of the real world that generated a potentially unfair decision, whereas we study the effect of sensitive information on a known classifier.

There exist several approaches to learning fair naive Bayes models. First, one may modify the data to achieve fairness and use standard algorithms to learn a classifier from the modified data. For instance, Kamiran and Calders (2009) proposed to change the labels for features near the decision boundary to achieve statistical parity, while the repair algorithm of Feldman et al. (2015) changes the non-sensitive attributes to reduce their correlation with the sensitive attribute. Although these methods have the flexibility of learning different models, we have shown empirically that a model learned from a fair data may still exhibit discrimination patterns. On the other hand, Calders and Verwer (2010) proposed three different Bayesian network structures modified from a naive Bayes network in order to enforce statistical parity directly during learning. We have shown in the previous section that our method achieves better accuracy than their two-naive-Bayes method (which was found to be the best of three methods), while ensuring a stricter definition of fairness. Lastly, one may add a regularizer during learning (Kamishima et al. 2012; Zemel et al. 2013), whereas we formulated to problem as constrained optimization, an approach often used to ensure fairness in other models (Dwork et al. 2012; Kearns et al. 2018).

6 Discussion and Conclusion

In this paper we introduced a novel definition of fair probability distribution in terms of discrimination patterns which considers exponentially many (partial) observations of features. We have also presented algorithms to search for discrimination patterns in naive Bayes networks and to learn a high-quality fair naive Bayes classifier from data. We empirically demonstrated the efficiency of our search algorithm and the ability to eliminate exponentially many discrimination patterns by iteratively removing a small fraction at a time.

We have shown that our approach of fair distribution implies group fairness such as statistical parity. However, ensuring group fairness in general is always with respect to a distribution and is only valid under the assumption that this distribution is truthful. While our approach guarantees some level of group fairness of naive Bayes classifiers, this is only true if the naive Bayes assumption holds. That is, the group fairness guarantees do not extend to using the classifier on an arbitrary population.

There is always a tension between three criteria of a probabilistic model: its fidelity, fairness, and tractability. Our approach aims to strike a balance between them by giving up some likelihood to be tractable (naive Bayes assumption) and more fair. There are certainly other valid approaches: learning a more general graphical model to increase fairness and truthfulness, which would in general make it intractable, or making the model less fair in order to make it more truthful and tractable.

Lastly, real-world algorithmic fairness problems are only solved by domain experts understanding the process that generated the data, its inherent biases, and which modeling assumptions are appropriate. Our algorithm is only a tool to assist such experts in learning fair distributions: it can provide the domain expert with discrimination patterns, who can then decide which patterns need to be eliminated.

Acknowledgments This work is partially supported by NSF grants #IIS-1633857, #CCF-1837129, DARPA XAI grant #N66001-17-2-4032, NEC Research, and gifts from Intel and Facebook Research. Golnoosh Farnadi and Behrouz Babaki are supported by postdoctoral scholarships from IVADO through the Canada First Research Excellence Fund (CFREF) grant.

References

Calders, T., and Verwer, S. 2010. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21(2):277–292.

Chouldechova, A. 2017. Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. *Big data* 5(2):153–163.

Darwiche, A. 2009. *Modeling and reasoning with Bayesian networks*. Cambridge University Press.

Datta, A.; Tschantz, M. C.; and Datta, A. 2015. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies* 2015(1):92–112.

Dechter, R. 2013. Reasoning with probabilistic and deterministic graphical models: Exact algorithms. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 7(3):1–191.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226. ACM.

Ecker, J. G. 1980. Geometric programming: methods, computations and applications. *SIAM review* 22(3):338–362.

Farnadi, G.; Babaki, B.; and Getoor, L. 2018. Fairness in relational domains. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 108–114. ACM.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268. ACM.

Hardt, M.; Price, E.; Srebro, N.; et al. 2016. Equality of opportunity in supervised learning. In *NeurIPS*, 3315–3323.

Henderson, L.; Herring, C.; Horton, H. D.; and Thomas, M. 2015. Credit where credit is due?: Race, gender, and discrimination in the credit scores of business startups. *The Review of Black Political Economy* 42(4):459–479.

Kamiran, F., and Calders, T. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, 1–6. IEEE.

Kamishima, T.; Akaho, S.; Asoh, H.; and Sakuma, J. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 35–50. Springer.

Kearns, M.; Neel, S.; Roth, A.; and Wu, Z. S. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, 2564–2572.

Khosravi, P.; Liang, Y.; Choi, Y.; and Van den Broeck, G. 2019. What to expect of classifiers? reasoning about logistic regression with missing features. In *IJCAI*.

Kilbertus, N.; Carulla, M. R.; Parascandolo, G.; Hardt, M.; Janzing, D.; and Schölkopf, B. 2017. Avoiding discrimination through causal reasoning. In *NeurIPS*, 656–666.

Kisa, D.; Van den Broeck, G.; Choi, A.; and Darwiche, A. 2014. Probabilistic sentential decision diagrams. In *KR*.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual fairness. In *NeurIPS*, 4066–4076.

Larson, J.; Mattu, S.; Kirchner, L.; and Angwin, J. 2016. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)* 9.

Luong, B. T.; Ruggieri, S.; and Turini, F. 2011. 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*, 502–510. ACM.

Mohler, G.; Raje, R.; Carter, J.; Valasik, M.; and Brantingham, J. 2018. A penalized likelihood method for balancing accuracy and fairness in predictive policing. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2454–2459. IEEE.

Poon, H., and Domingos, P. 2011. Sum-product networks: a new deep architecture. In *UAI*, 337–346.

Rahman, T.; Kothalkar, P.; and Gogate, V. 2014. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *ECML/PKDD*.

Russell, C.; Kusner, M. J.; Loftus, J.; and Silva, R. 2017. When worlds collide: integrating different counterfactual assumptions in fairness. In *NeurIPS*, 6414–6423.

Salimi, B.; Rodriguez, L.; Howe, B.; and Suciu, D. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, 793–810. ACM.

Tramer, F.; Atlidakis, V.; Geambasu, R.; Hsu, D.; Hubaux, J.-P.; Humbert, M.; Juels, A.; and Lin, H. 2017. Fairtest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 401–416. IEEE.

Zafar, M. B.; Valera, I.; Gomez-Rodriguez, M.; Gummadi, K. P.; and Weller, A. 2017a. From parity to preference-based notions of fairness in classification. In *NeurIPS*, 228–238.

Zafar, M. B.; Valera, I.; Gomez Rodriguez, M.; and Gummadi, K. P. 2017b. Fairness constraints: Mechanisms for fair classification. In *20th International Conference on Artificial Intelligence and Statistics*, 962–970.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *ICML*, 325–333.