Learning Fair Naive Bayes Classifiers by Discovering and Eliminating Discrimination Patterns

YooJung Choi*
University of California, Los Angeles
yjchoi@cs.ucla.edu

Golnoosh Farnadi*
Mila & Université de Montréal
farnadig@mila.quebec

Behrouz Babaki*
Polytechnique Montréal
behrouz.babaki@polymtl.ca

Guy Van den Broeck
University of California, Los Angeles
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 [11], recidivism prediction [2], insurance pricing [10], hiring [3], and credit rating [8]. The algorithmic fairness literature has proposed various solutions, including individual fairness [4, 13], statistical parity and group fairness [2, 6, 9], counterfactual fairness [10], preference-based fairness [12], and equality of opportunity [7]. 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 a decision variable. In particular, Bayesian network classifiers treat the classification task as a probabilistic inference problem: given observed features, compute the probability of the decision variable. Such models can effectively handle missing features at prediction time by simply marginalizing the unobserved variables out of the distribution. 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, with two proposed criteria for identifying

*Equal contribution

NeurIPS 2019 Workshop on Machine Learning with Guarantees, Vancouver, Canada.

the most important discrimination patterns. We specialize our pattern miner to effectively discover discrimination patterns in naive Bayes models using branch-and-bound search, exploiting the naive Bayes assumption for efficient computation of bounds during search. The second key contribution of this paper is a parameter learning algorithm for naive Bayes classifiers that eliminates discrimination patterns from the learned distribution. We propose a signomial programming approach to eliminate discrimination patterns during maximum-likelihood learning. Moreover, to efficiently eliminate an exponential number of possible patterns, we propose a cutting-plane approach that iteratively finds and eliminates discrimination patterns until the entire learned model is fair. 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. Also, we empirically demonstrate that our iterative learning converges in a small number of iterations, while effectively removing millions of discrimination patterns. Moreover, the learned fair models are of high quality, achieving likelihoods that are close to that of unfair max-likelihood models, as well as accuracy higher 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: $\Delta_{P,d}(\mathbf{x}, \mathbf{y}) \triangleq P(d \mid \mathbf{xy}) - P(d \mid \mathbf{y})$.*

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.

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, for all individuals and subgroups. Hence, 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

Verifying δ-fairness A naive way to check δ-fairness is to 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. Instead, we use branch-and-bound search to more efficiently decide if a model is fair. Our algorithm recursively adds variable instantiations and checks the discrimination score at each step. If the input distribution is δ-fair, it returns no pattern; otherwise, it returns the set of all discriminating patterns. Furthermore, we propose the following bound for the discrimination score to prune the search tree and avoid enumerating all patterns.

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 $\mathbf{Y}' = \mathbf{Z} \setminus (\mathbf{XY})$. Let $\widetilde{\Delta}(\alpha, \beta, \gamma) \triangleq \frac{\alpha\gamma}{\alpha\gamma + \beta(1-\gamma)} - \gamma$. Then the degrees of discrimination for all patterns*

$\mathbf{xx'yy'}$ *that extend* $\mathbf{xy}$ *are bounded as follows:*

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

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, allowing us to efficiently compute the bounds as closed-form solutions. We refer to the Appendix for full proofs and details of the algorithm.

Searching for top-k ranked patterns If a distribution is significantly unfair, our verification algorithm may return exponentially many patterns. This is not only very expensive but also difficult to interpret. Instead, we would like to return a smaller set of "interesting" discrimination patterns, thus calling for a ranking among patterns. An obvious choice is to look at those with the highest discrimination scores. Searching for the k most discriminating patterns can be done by simply modifying the verification algorithm to keep track of only the top-k patterns at each point in search.

Nevertheless, ranking patterns by their discrimination score may return those of extremely low probability, which may be of lesser interest as the probability of a discrimination pattern denotes the proportion of the population (according to the distribution) that could be affected unfairly. To address this, we propose a more sophisticated ranking 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:*

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

The divergence score assigns to a pattern $\mathbf{xy}$ the minimum Kullback-Leibler 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. To find the top-k patterns with the divergence score, we need to be able to compute the score and its upper bound efficiently. They can in fact be computed in linear time for naive Bayes classifiers; we refer to the appendix for details.

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. It is based on formulating the learning subject to fairness constraints as a signomial program, an optimization problem that minimizes a signomial objective function subject to signomial inequality and monomial equality constraints [5].

Parameter learning with fairness constraints Given data $\mathcal{D}$, we learn the maximum-likelihood parameters by minimizing the inverse of the likelihood $\prod_i \theta_i^{-n_i}$ where n_i is the number of examples in $\mathcal{D}$ that satisfy the assignment corresponding to parameter θ_i. The parameters of a naive Bayes network with binary class consist of $\theta_d, \theta_{\bar{d}}$, and $\theta_{z\mid d}, \theta_{z\mid \bar{d}}$ for all z. To learn a valid distribution, we add constraints to ensure that probabilities are non-negative and sum to one.[2] Lastly, to ensure that a pattern $\mathbf{xy}$ is non-discriminating, we add the following constraints to our optimization problem.

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 $|\Delta_{P_\theta,d}(\mathbf{x},\mathbf{y})| \le \delta$ for a threshold $\delta \in [0,1]$ iff the following holds:*

$$r_{\mathbf{x}} = \frac{\prod_x \theta_{x\mid\bar{d}}}{\prod_x \theta_{x\mid d}}, \qquad r_{\mathbf{y}} = \frac{\theta_{\bar{d}} \prod_y \theta_{y\mid\bar{d}}}{\theta_d \prod_y \theta_{y\mid d}},$$

$$\left(\frac{1-\delta}{\delta}\right) r_{\mathbf{x}} r_{\mathbf{y}} - \left(\frac{1+\delta}{\delta}\right) r_{\mathbf{y}} - r_{\mathbf{x}} r_{\mathbf{y}}^2 \le 1, \qquad -\left(\frac{1+\delta}{\delta}\right) r_{\mathbf{x}} r_{\mathbf{y}} + \left(\frac{1-\delta}{\delta}\right) r_{\mathbf{y}} - r_{\mathbf{x}} r_{\mathbf{y}}^2 \le 1.$$

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. We use *GPkit* to find local solutions to these problems.

[2]The former is inherent to signomial programs; to enforce the latter, for each d and feature Z we add signomial inequality constraints: $\sum_z \theta_{z\mid d} \le 1$ and $2 - \sum_z \theta_{z\mid d} \le 1$.

Table 1: Log-likelihood of models learned without fairness constraints, with the δ-fair learner, and with decision-independent sensitive variables.

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: Accuracy of δ-fair models, two-naive-Bayes method, and naive Bayes models trained on repaired, discrimination-free data.

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

Iterative learning Learning a model that is entirely fair with this approach will introduce an exponential number of constraints. To address this challenge, 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. 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 search algorithm finds no more discrimination pattern. While our algorithm could in the worst case add exponentially many constraints, we will later show empirically that we can learn a δ-fair model by explicitly enforcing only a small fraction of fairness constraints.

5 Empirical evaluation

We empirically evaluate our discrimination pattern miner and δ-fair learning on *COMPAS*, *Adult* and *German* datasets, used for predicting recidivism, income level, and credit risk respectively.[3] A summary of the datasets and the full set of results can be found in the Appendix.

Discrimination pattern miner To see the efficiency of our pattern miner, we inspect the fraction of all possible patterns that our algorithm visits during search. We evaluate on three datasets, using two rank heuristics (discrimination and divergence), three δ values (0.01, 0.05, and 0.1), and three k values (1, 10, and 100). When mining for the top-k patterns, our algorithm visited as few as 7.5e-8 of all possible patterns, indicating that it prunes large parts of the search space. It explored more than 10% of the search space in only 9 out of 54 instances, and only for the COMPAS dataset which has a much smaller search space (15K as opposed to 23B for German dataset).

δ-fair learner Next, we evaluate the effectiveness of our iterative δ-fair learner, again on three datasets, two ranking heuristics, and varying k and δ values. We observed that enforcing a small number of (even a single) fairness constraints can eliminate a large number of remaining discrimination patterns. In particular, on COMPAS dataset with $k = 1$ and $\delta = 0.1$, adding only the most discriminating pattern as a constraint at each iteration produced an entirely δ-fair model with only three iterations, eliminating all 2695 discrimination patterns of the unconstrained naive Bayes model. On the other datasets, more than a million discrimination patterns were eliminated using a few dozen to, even in the worst case, a few thousand fairness constraints. Furthermore, stricter fairness requirements (smaller δ) tended to require more iterations, as would be expected. Interestingly, neither of the rankings consistently dominated the other in terms of the number of iterations to converge.

Lastly, we compare the quality of naive Bayes models from our fair learner in terms of log-likelihoods as well as accuracy. We first compare against (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, with max-likelihood learning for the remaining parameters (independent). These models lie at two opposite ends of the spectrum of the trade-off between fairness and predictive power, and the δ-fair model falls between these extremes. As shown in Table 1, the δ-fair models achieve likelihoods that are much closer to those of the unconstrained models than the independent ones, demonstrating that it is possible to enforce fairness without a major reduction in model quality. Table 2 reports the 10-fold CV accuracy of our method (δ-fair) compared to the unconstrained NB model and two other methods of learning fair classifiers: the two-naive-Bayes method (2NB) [1], and a naive Bayes model trained on discrimination-free data using the repair algorithm of Feldman et al. [6] 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 with fully-observed features.

[3] `https://github.com/propublica/compas-analysis` and `https://archive.ics.uci.edu/ml`

References

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

[2] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *CoRR*, abs/1703.00056, 2017.

[3] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015.

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

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

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

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

[8] Loren Henderson, Cedric Herring, Hayward Derrick Horton, and Melvin Thomas. 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, 2015.

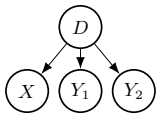
[9] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[10] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *NIPS*, pages 4069–4079, 2017.

[11] George Mohler, Rajeev Raje, Jeremy Carter, Matthew Valasik, and Jeffrey Brantingham. A penalized likelihood method for balancing accuracy and fairness in predictive policing. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2454–2459. IEEE, 2018.

[12] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, Krishna P. Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. In *NIPS*, pages 228–238, 2017.

[13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.



$P(d)$
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

A Intuition for discrimination patterns

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 where 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 $\max_{xy_1y_2} |\Delta(x, y_1y_2)| = 0.167 \leq \delta$, with the highest discrimination score achieved at $\Delta(\bar{x}, y_1\bar{y_2}) = -0.167$.

Even though the example network has no discrimination pattern 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. Therefore, we define δ-fairness to ensure that there exists no discrimination pattern across all subsets of observable features.

B δ-Fairness Verification Algorithm

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$(\mathbf{x}, \mathbf{y}, \mathbf{E} \cup \{Z\})$

Algorithm 1 finds discrimination patterns. It recursively adds variable instantiations and checks the discrimination score at each step. Specifically, $\mathbf{x}$ and $\mathbf{y}$ denote the pattern that has been constructed until the current point in recursion, and $\mathbf{E}$ contain variables that should be excluded in the following search steps. Furthermore, UB$(\mathbf{x}, \mathbf{y}, \mathbf{E})$ bounds the degree of discrimination achievable by observing more features after $\mathbf{xy}$ while excluding features $\mathbf{E}$. At the end of search, if the input distribution is δ-fair, the algorithm returns no pattern; otherwise, it returns the set of all discriminating patterns.

C Degree of Discrimination Bound

We now prove the correctness of our bound on discrimination score, and describe how it can be computed efficiently for naive Bayes networks.

C.1 Proof of Proposition 1

We first derive how $\widetilde{\Delta}$ represents the degree of discrimination Δ for some pattern $\mathbf{xy}$.

$$\begin{aligned}\Delta_{P,d}(\mathbf{x}, \mathbf{y}) &= P(d \,|\, \mathbf{xy}) - P(d \,|\, \mathbf{y}) \\ &= \frac{P(\mathbf{x} \,|\, d)P(d\mathbf{y})}{P(\mathbf{x} \,|\, d)P(d\mathbf{y}) + P(\mathbf{x} \,|\, \overline{d})P(\overline{d}\mathbf{y})} - P(d \,|\, \mathbf{y})\end{aligned}$$

$$= \frac{P(\mathbf{x} \mid d)P(d \mid \mathbf{y})}{P(\mathbf{x} \mid d)P(d \mid \mathbf{y}) + P(\mathbf{x} \mid \overline{d})P(\overline{d} \mid \mathbf{y})} - P(d \mid \mathbf{y})$$
$$= \widetilde{\Delta}(P(\mathbf{x} \mid d), P(\mathbf{x} \mid \overline{d}), P(d \mid \mathbf{y}))$$

Clearly, if $l \leq \gamma \leq u$ then $\min_{l \leq \gamma \leq u} \widetilde{\Delta}(\alpha, \beta, \gamma) \leq \widetilde{\Delta}(\alpha, \beta, \gamma) \leq \max_{l \leq \gamma \leq u} \widetilde{\Delta}(\alpha, \beta, \gamma)$. Therefore, if $l \leq P(d \mid \mathbf{y}\mathbf{y}') \leq u$, then the following holds for any $\mathbf{x}$:

$$\min_{l \leq \gamma \leq u} \widetilde{\Delta}(P(\mathbf{x} \mid d), P(\mathbf{x} \mid \overline{d}), \gamma) \leq \widetilde{\Delta}(P(\mathbf{x} \mid d), P(\mathbf{x} \mid \overline{d}), P(d \mid \mathbf{y}\mathbf{y}'))$$
$$= \Delta_{P,d}(\mathbf{x}, \mathbf{y}\mathbf{y}') \leq \max_{l \leq \gamma \leq u} \widetilde{\Delta}(P(\mathbf{x} \mid d), P(\mathbf{x} \mid \overline{d}), \gamma).$$

Next, suppose $\mathbf{x}'_u = \arg\max_{\mathbf{x}'} P(d \mid \mathbf{x}\mathbf{x}')$ and $\mathbf{x}'_l = \arg\min_{\mathbf{x}'} P(d \mid \mathbf{x}\mathbf{x}')$. Then from Lemma 1, we also have that $\mathbf{x}'_u = \arg\max_{\mathbf{x}'} P(d \mid \mathbf{x}\mathbf{x}'\mathbf{y}\mathbf{y}')$ and $\mathbf{x}'_l = \arg\min_{\mathbf{x}'} P(d \mid \mathbf{x}\mathbf{x}'\mathbf{y}\mathbf{y}')$ for any $\mathbf{y}\mathbf{y}'$. Therefore,

$$\min_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_l \mid d), P(\mathbf{x}\mathbf{x}'_l \mid \overline{d}), \gamma\right)$$
$$\leq \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_l \mid d), P(\mathbf{x}\mathbf{x}'_l \mid \overline{d}), P(d \mid \mathbf{y}\mathbf{y}')\right) = \Delta_{P,d}(\mathbf{x}\mathbf{x}'_l, \mathbf{y}\mathbf{y}') = P(d \mid \mathbf{x}\mathbf{x}'_l\mathbf{y}\mathbf{y}') - P(d \mid \mathbf{y}\mathbf{y}')$$
$$\leq P(d \mid \mathbf{x}\mathbf{x}'\mathbf{y}\mathbf{y}') - P(d \mid \mathbf{y}\mathbf{y}') = \Delta_{P,d}(\mathbf{x}\mathbf{x}', \mathbf{y}\mathbf{y}')$$
$$\leq \Delta_{P,d}(\mathbf{x}\mathbf{x}'_u, \mathbf{y}\mathbf{y}') = \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_u \mid d), P(\mathbf{x}\mathbf{x}'_u \mid \overline{d}), P(d \mid \mathbf{y}\mathbf{y}')\right)$$
$$\leq \max_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{x}\mathbf{x}'_u \mid d), P(\mathbf{x}\mathbf{x}'_u \mid \overline{d}), \gamma\right). \qquad \square$$

C.2 Computing the Discrimination Bound

To apply above proposition, we need to find $\mathbf{x}'_u, \mathbf{x}'_l, l, u$ by maximizing/minimizing $P(d|\mathbf{x}\mathbf{x}')$ and $P(d|\mathbf{y}\mathbf{y}')$ for a given pattern $\mathbf{x}\mathbf{y}$. Note that above proposition can always be applied using $l = 0$ and $u = 1$, in which case the bounds are simply the smallest and largest degrees of discrimination achievable from current $\mathbf{x}$, regardless of current $\mathbf{y}$. However, values of l and u specific to the current pattern will tighten the bounds and improve pruning. Fortunately, we can efficiently compute them 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:* $\arg\max_{\mathbf{v}} P(d|\mathbf{v}\mathbf{w}) = \left\{\arg\max_{v_i} P(v_i|d)/P(v_i|\overline{d})\right\}_{i=1}^n$.

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. The proof is as follows.

Proof. It suffices to prove that for a single variable V and all evidence $\mathbf{w}$, $\arg\max_v P(d \mid v\mathbf{w}) = \arg\max_v \frac{P(v \mid d)}{P(v \mid \overline{d})}$. We first express $P(d \mid v\mathbf{w})$ as the following:

$$P(d \mid v\mathbf{w}) = \frac{P(v \mid d)P(d \mid \mathbf{w})}{P(v \mid d)P(d \mid \mathbf{w}) + P(v \mid \overline{d})P(\overline{d} \mid \mathbf{w})} = \frac{1}{1 + \frac{P(v \mid \overline{d})P(\overline{d} \mid \mathbf{w})}{P(v \mid d)P(d \mid \mathbf{w})}}$$

Then clearly,

$$\arg\max_v P(d \mid v\mathbf{w}) = \arg\min_v \frac{P(v \mid \overline{d})P(\overline{d} \mid \mathbf{w})}{P(v \mid d)P(d \mid \mathbf{w})} = \arg\max_v \frac{P(v \mid d)}{P(v \mid \overline{d})}.$$

$\square$

Given the parameters for Proposition 1, we can compute the upper bound if we can efficiently optimize $\widetilde{\Delta}(\alpha, \beta, \gamma)$ over $l \leq \gamma \leq u$ for a fixed α, β.

If $\alpha = 0$ and $\beta = 0$, then the probability of the pattern is zero and thus the conditional probability of the decision variable is ill-defined. Therefore, we will assume that either α or β is nonzero. Let us write $\widetilde{\Delta}_{\alpha,\beta}(\gamma) = \widetilde{\Delta}(\alpha, \beta, \gamma)$ to denote the function restricted to fixed α and β. If $\alpha = \beta$, then

$\widetilde{\Delta}_{\alpha,\beta} = 0$. Also, $\widetilde{\Delta}_{0,\beta}(\gamma) = -\gamma$ and $\widetilde{\Delta}_{\alpha,0}(\gamma) = 1 - \gamma$. Thus, in the following analysis we assume α and β are non-zero and distinct.

If $0 < \alpha \leq \beta \leq 1$, $\widetilde{\Delta}_{\alpha,\beta}$ is negative and convex in γ within $0 \leq \gamma \leq 1$. On the other hand, if $0 < \beta \leq \alpha \leq 1$, then $\widetilde{\Delta}_{\alpha,\beta,\gamma}$ is positive and concave. This can quickly be checked using the following derivatives.

$$\frac{d}{d\gamma}\widetilde{\Delta}_{\alpha,\beta}(\gamma) = \frac{\alpha\beta}{(\alpha\gamma + \beta(1-\gamma))^2} - 1, \quad \frac{d^2}{d\gamma^2}\widetilde{\Delta}_{\alpha,\beta}(\gamma) = \frac{-2\alpha\beta(\alpha-\beta)}{(\alpha\gamma + \beta(1-\gamma))^3}$$

Furthermore, the sign of the derivative at $\gamma = 0$ is different from that at $\gamma = 1$, and thus there must exist a unique optimum in $0 \leq \gamma \leq 1$.

Solving for $\frac{d}{d\gamma}\widetilde{\Delta}_{\alpha,\beta}(\gamma) = 0$, we get $\gamma = \frac{\beta \pm \sqrt{\alpha\beta}}{\beta - \alpha}$. The solution corresponding to the feasible space $0 \leq \gamma \leq 1$ is: $\gamma_{\text{opt}} = \frac{\beta - \sqrt{\alpha\beta}}{\beta - \alpha}$. The optimal value is derived as the following.

$$\widetilde{\Delta}_{\alpha,\beta}(\gamma_{\text{opt}}) = \frac{\alpha\left(\frac{\beta-\sqrt{\alpha\beta}}{\beta-\alpha}\right)}{(\alpha-\beta)\left(\frac{\beta-\sqrt{\alpha\beta}}{\beta-\alpha}\right) + \beta} - \frac{\beta - \sqrt{\alpha\beta}}{\beta - \alpha} = \frac{\alpha(\beta - \sqrt{\alpha\beta})}{\sqrt{\alpha\beta}(\beta - \alpha)} - \frac{\beta - \sqrt{\alpha\beta}}{\beta - \alpha} = \frac{2\sqrt{\alpha\beta} - \alpha - \beta}{\beta - \alpha}$$

Then the optimal solution is: γ_{opt} if $l \leq \gamma_{\text{opt}} \leq u$; l if $\gamma_{\text{opt}} < l$; and u if $\gamma_{\text{opt}} > u$.

D Divergence Score

D.1 Computing the Divergence Score

We will show that the divergence score for a naive Bayes distribution can efficiently be computed as the following:

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

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 $\text{Div}_{P,d,\delta}(\mathbf{x},\mathbf{y}) = 0$ is attained when the pattern is already fair.

We want to find the closed form solution of the optimization problem in Equation 1. Because P and Q differs only in two assignments, we can write the KL divergence as follows:

$$\text{KL}(P \parallel Q) = \sum_{d\mathbf{z}} P(d\mathbf{z}) \log\left(\frac{P(d\mathbf{z})}{Q(d\mathbf{z})}\right) = P(d\mathbf{xy}) \log\left(\frac{P(d\mathbf{xy})}{Q(d\mathbf{xy})}\right) + P(\bar{d}\mathbf{xy}) \log\left(\frac{P(\bar{d}\mathbf{xy})}{Q(\bar{d}\mathbf{xy})}\right)$$

Let r be the change in probability of $d\mathbf{xy}$. That is, $r = Q(d\mathbf{xy}) - P(d\mathbf{xy})$. For Q to be a valid probability distribution, we must have $Q(d\mathbf{xy}) + Q(\bar{d}\mathbf{xy}) = P(\mathbf{xy})$. Then we have $Q(d\mathbf{xy}) = P(d\mathbf{xy}) + r$, and $Q(\bar{d}\mathbf{xy}) = P(\mathbf{xy}) - Q(d\mathbf{xy}) = P(\bar{d}\mathbf{xy}) - r$. We can then express the KL divergence between P and Q as a function of P and r:

$$g_{P,d,\mathbf{x},\mathbf{y}}(r) \triangleq P(d\mathbf{xy}) \log\left(\frac{P(d\mathbf{xy})}{P(d\mathbf{xy}) + r}\right) + P(\bar{d}\mathbf{xy}) \log\left(\frac{P(\bar{d}\mathbf{xy})}{P(\bar{d}\mathbf{xy}) - r}\right)$$

Moreover, the discrimination score of pattern $\mathbf{xy}$ w.r.t Q can be expressed using P and r as the following:

$$\begin{aligned} Q(d \mid \mathbf{xy}) - Q(d \mid \mathbf{y}) &= \frac{P(d\mathbf{xy}) + r}{P(\mathbf{xy})} - \frac{P(d\mathbf{y}) + r}{P(\mathbf{y})} = P(d \mid \mathbf{xy}) - P(d \mid \mathbf{y}) + r\left(\frac{1}{P(\mathbf{xy})} - \frac{1}{P(\mathbf{y})}\right) \\ &= \Delta_{P,d}(\mathbf{x},\mathbf{y}) + r\left(\frac{1}{P(\mathbf{xy})} - \frac{1}{P(\mathbf{y})}\right). \end{aligned}$$

The heuristic $\mathrm{Div}_{P,d,\delta}(\mathbf{x},\mathbf{y})$ is then written using r as follows:

$$\min_r \; g_{P,d,\mathbf{x},\mathbf{y}}(r) \;\; \text{s.t.} \;\; \left|\Delta_{P,d}(\mathbf{x},\mathbf{y}) + r\left(\frac{1}{P(\mathbf{xy})} - \frac{1}{P(\mathbf{y})}\right)\right| \leq \delta \qquad (3)$$
$$-P(d\mathbf{xy}) \leq r \leq P(\overline{d}\mathbf{xy})$$

The objective function $g_{P,d,\mathbf{x},\mathbf{y}}$ is convex in r with its unconstrained global minimum at $r = 0$. Note that this is a feasible point if and only if $|\Delta_{P,d}(\mathbf{x},\mathbf{y})| \leq \delta$; in other words, when the pattern $\mathbf{xy}$ is already fair. Otherwise, the optimum must be either of the extreme points of the feasible space, whichever is closer to 0. The extreme points for the first set of inequalities are:

$$r_1 = \frac{\delta - P(d\,|\,\mathbf{xy}) + P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}, \quad r_2 = \frac{-\delta - P(d\,|\,\mathbf{xy}) + P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}.$$

If $\Delta_{P,d}(\mathbf{x},\mathbf{y}) > \delta$, then $r_2 \leq r_1 < 0$. In such case, $g(r_2) \geq g(r_1)$ and $-P(d\mathbf{xy}) \leq r_1 \leq P(\overline{d}\mathbf{xy})$ as shown below:

$$r_1 < 0 \leq P(\overline{d}\mathbf{xy}),$$
$$-r_1 = \frac{-\delta + P(d\,|\,\mathbf{xy}) - P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} \leq \frac{P(d\,|\,\mathbf{xy}) - P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} \leq \frac{P(d\,|\,\mathbf{xy}) - P(d\mathbf{x}\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} = P(d\mathbf{xy})$$

Similarly, if $\Delta_{P,d}(\mathbf{x},\mathbf{y}) < -\delta$, then $r_1 \geq r_2 > 0$. Also, $g(r_1) \geq g(r_2)$ and $-P(d\mathbf{xy}) \leq r_2 \leq P(\overline{d}\mathbf{xy})$ as shown below:

$$r_2 > 0 \geq -P(d\mathbf{xy}),$$
$$r_2 \leq \frac{-P(d\,|\,\mathbf{xy}) + P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} \leq \frac{P(\overline{d}\,|\,\mathbf{xy}) - P(\overline{d}\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} = P(\overline{d}\mathbf{xy})$$

Hence, the optimal solution $r^\star$ is

$$r^\star = \begin{cases} 0, & \text{if } |\Delta_{P,d}(\mathbf{x},\mathbf{y})| \leq \delta, \\ \frac{\delta - \Delta_{P,d}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}, & \text{if } \Delta_{P,d}(\mathbf{x},\mathbf{y}) > \delta, \\ \frac{-\delta - \Delta_{P,d}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}, & \text{if } \Delta_{P,d}(\mathbf{x},\mathbf{y}) < -\delta, \end{cases}$$

and the divergence score is $\mathrm{Div}_{P,d,\delta}(\mathbf{x},\mathbf{y}) = g_{P,d,\mathbf{x},\mathbf{y}}(r^\star)$.

D.2 Upper Bounds on Divergence Score

Here we present two upper bounds on the divergence score for pruning the search tree. The first bound uses the observation that the hypothetical distribution Q with $\Delta_{Q,d}(\mathbf{x},\mathbf{y}) = 0$ is always a feasible hypothetical fair distribution.

Proposition 3. *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 $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$. For all possible valid extensions $\mathbf{x}'$ and $\mathbf{y}'$, the following holds:*

$$\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') \leq \; P(d\mathbf{xy}) \log \frac{\max_{\mathbf{z} \models \mathbf{xy}} P(d\,|\,\mathbf{z})}{\min_{\mathbf{z} \models \mathbf{y}} P(d\,|\,\mathbf{z})} + P(\overline{d}\mathbf{xy}) \log \frac{\max_{\mathbf{z} \models \mathbf{xy}} P(\overline{d}\,|\,\mathbf{z})}{\min_{\mathbf{z} \models \mathbf{y}} P(\overline{d}\,|\,\mathbf{z})}$$

Proof. Consider the following point:

$$r_0 = \frac{-P(d\,|\,\mathbf{xy}) + P(d\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})}.$$

First, we show that above r_0 is always a feasible point in Problem 3:

$$\left|\Delta_{P,d}(\mathbf{x},\mathbf{y}) + r_0\left(\frac{1}{P(\mathbf{xy})} - \frac{1}{P(\mathbf{y})}\right)\right| = |\Delta_{P,d}(\mathbf{x},\mathbf{y}) - \Delta_{P,d}(\mathbf{x},\mathbf{y})| = 0 \leq \delta,$$

$$r_0 = \frac{P(\overline{d}\,|\,\mathbf{xy}) - P(\overline{d}\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} \leq \frac{P(\overline{d}\,|\,\mathbf{xy}) - P(\overline{d}\mathbf{x}\,|\,\mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} = P(\overline{d}\mathbf{xy}),$$

$$-r_0 = \frac{P(d \,|\, \mathbf{xy}) - P(d \,|\, \mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} \leq \frac{P(d \,|\, \mathbf{xy}) - P(d\mathbf{x} \,|\, \mathbf{y})}{1/P(\mathbf{xy}) - 1/P(\mathbf{y})} = P(d\mathbf{xy}).$$

Then the divergence score for any pattern must be smaller than $g_{P,d,\mathbf{x},\mathbf{y}}(r_0)$:

$$\begin{aligned}\mathrm{Div}_{P,d,\delta}(\mathbf{x},\mathbf{y}) &\leq g_{P,d,\mathbf{x},\mathbf{y}}(r_0) = P(d\mathbf{xy}) \log \frac{P(d \,|\, \mathbf{xy})}{P(d \,|\, \overline{\mathbf{x}}\mathbf{y})} + P(\overline{d}\mathbf{xy}) \log \frac{P(\overline{d} \,|\, \mathbf{xy})}{P(\overline{d} \,|\, \overline{\mathbf{x}}\mathbf{y})} \\ &\leq P(d\mathbf{xy}) \log \frac{P(d \,|\, \mathbf{xy})}{\min_{\mathbf{x}} P(d \,|\, \mathbf{xy})} + P(\overline{d}\mathbf{xy}) \log \frac{P(\overline{d} \,|\, \mathbf{xy})}{\min_{\mathbf{x}} P(\overline{d} \,|\, \mathbf{xy})}.\end{aligned}$$

Here, we use $\overline{\mathbf{x}}$ to mean that $\mathbf{x}$ does not hold. In other words,

$$P(d \,|\, \overline{\mathbf{x}}\mathbf{y}) = \frac{P(d\mathbf{y}) - P(d\mathbf{xy})}{P(\mathbf{y}) - P(\mathbf{xy})} = \sum_{\mathbf{x}} P(d \,|\, \mathbf{xy}) P(\mathbf{x} \,|\, \overline{\mathbf{x}}\mathbf{y}).$$

We can then use this to bound the divergence score any pattern extended from $\mathbf{xy}$:

$$\begin{aligned}&\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') \\ &\leq P(d\mathbf{xx}'\mathbf{yy}') \log \frac{P(d \,|\, \mathbf{xx}'\mathbf{yy}')}{\min_{\mathbf{xx}'} P(d \,|\, \mathbf{xx}'\mathbf{yy}')} + P(\overline{d}\mathbf{xx}'\mathbf{yy}') \log \frac{P(\overline{d} \,|\, \mathbf{xx}'\mathbf{yy}')}{\min_{\mathbf{xx}'} P(\overline{d} \,|\, \mathbf{xx}'\mathbf{yy}')} \\ &\leq P(d\mathbf{xy}) \log \frac{\max_{\mathbf{z} \models \mathbf{xy}} P(d \,|\, \mathbf{z})}{\min_{\mathbf{z} \models \mathbf{y}} P(d \,|\, \mathbf{z})} + P(\overline{d}\mathbf{xy}) \log \frac{\max_{\mathbf{z} \models \mathbf{xy}} P(\overline{d} \,|\, \mathbf{z})}{\min_{\mathbf{z} \models \mathbf{y}} P(\overline{d} \,|\, \mathbf{z})}.\end{aligned}$$

□

We can also bound the divergence score using the maximum and minimum possible discrimination scores shown in Proposition 1, in place of the current pattern's discrimination. Let us denote the bounds for discrimination score as follows:

$$\overline{\Delta}(\mathbf{x},\mathbf{y}) = \max_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{xx}'_u \,|\, d), P(\mathbf{xx}'_u \,|\, \overline{d}), \gamma\right), \; \underline{\Delta}(\mathbf{x},\mathbf{y}) = \min_{l \leq \gamma \leq u} \widetilde{\Delta}\left(P(\mathbf{xx}'_l \,|\, d), P(\mathbf{xx}'_l \,|\, \overline{d}), \gamma\right).$$

Proposition 4. *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 $\mathbf{Y} \subseteq \mathbf{Z} \setminus \mathbf{X}$. For all possible valid extensions $\mathbf{x}'$ and $\mathbf{y}'$, $\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') \leq \max\left(g_{P,d,\mathbf{xx}',\mathbf{yy}'}(r_u), g_{P,d,\mathbf{xx}',\mathbf{yy}'}(r_l)\right)$ where*

$$r_u = \frac{\delta - \overline{\Delta}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')}, \quad r_l = \frac{-\delta - \underline{\Delta}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')}.$$

Proof. The proof proceeds by case analysis on the discrimination score of extended patterns $\mathbf{xx}'\mathbf{yy}'$.

First, if $|\Delta(\mathbf{xx}', \mathbf{yy}')| \leq \delta$, $\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') = 0$ which is the global minimum, and thus is smaller than both $g(r_u)$ and $g(r_l)$.

Next, suppose $\Delta(\mathbf{xx}', \mathbf{yy}') > \delta$. Then from Proposition 1,

$$r_u = \frac{\delta - \overline{\Delta}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')} \leq r^\star = \frac{\delta - \Delta_{P,d}(\mathbf{xx}', \mathbf{yy}')}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')} < 0.$$

As g is convex with its minimum at 0, we can conclude $\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') = g(r^\star) \leq g(r_u)$.

Finally, if $\Delta(\mathbf{xx}', \mathbf{yy}') < -\delta$, we have

$$r_l = \frac{-\delta - \underline{\Delta}(\mathbf{x},\mathbf{y})}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')} \geq r^\star = \frac{-\delta - \Delta_{P,d}(\mathbf{xx}', \mathbf{yy}')}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')} > 0.$$

Similarly, this implies $\mathrm{Div}_{P,d,\delta}(\mathbf{xx}', \mathbf{yy}') = g(r^\star) \leq g(r_l)$. Because the divergence score is always smaller than either $g(r_u)$ or $g(r_l)$, it must be smaller than $\max(g(r_u), g(r_l))$. □

Lastly, we show how to efficiently compute an upper bound on $g_{P,d,\mathbf{xx}',\mathbf{yy}'}(r_u)$ $g_{P,d,\mathbf{xx}',\mathbf{yy}'}(r_l)$ from Proposition 4 for all patterns extended from $\mathbf{xy}$. This is necessary for pruning during the search for discrimination patterns with high divergence scores. First, note that r_u and r_l can be expressed as

$$\frac{c}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')}, \tag{4}$$

where $c = \delta - \overline{\Delta}(\mathbf{x},\mathbf{y})$ for r_u and $c = -\delta - \underline{\Delta}(\mathbf{x},\mathbf{y})$ for r_l. Hence, it suffices to derive the following bound.

$$\begin{aligned}
&g_{P,d,\mathbf{xx}',\mathbf{yy}'}\left(\frac{c}{1/P(\mathbf{xx}'\mathbf{yy}') - 1/P(\mathbf{yy}')}\right) \\
&= P(d\mathbf{xx}'\mathbf{yy}')\log\left(\frac{P(d\mathbf{xx}'\mathbf{yy}')}{P(d\mathbf{xx}'\mathbf{yy}') + \frac{c}{1/P(\mathbf{xx}'\mathbf{yy}')-1/P(\mathbf{yy}')}}\right) \\
&\quad + P(\bar{d}\mathbf{xx}'\mathbf{yy}')\log\left(\frac{P(\bar{d}\mathbf{xx}'\mathbf{yy}')}{P(\bar{d}\mathbf{xx}'\mathbf{yy}') - \frac{c}{1/P(\mathbf{xx}'\mathbf{yy}')-1/P(\mathbf{yy}')}}\right) \\
&= P(d\mathbf{xx}'\mathbf{yy}')\log\left(\frac{P(d\,|\,\mathbf{xx}'\mathbf{yy}')(1-P(\mathbf{xx}'\,|\,\mathbf{yy}'))}{P(d\,|\,\mathbf{xx}'\mathbf{yy}')(1-P(\mathbf{xx}'\,|\,\mathbf{yy}')) + c}\right) \\
&\quad + P(\bar{d}\mathbf{xx}'\mathbf{yy}')\log\left(\frac{P(\bar{d}\,|\,\mathbf{xx}'\mathbf{yy}')(1-P(\mathbf{xx}'\,|\,\mathbf{yy}'))}{P(\bar{d}\,|\,\mathbf{xx}'\mathbf{yy}')(1-P(\mathbf{xx}'\,|\,\mathbf{yy}')) - c}\right) \\
&\leq \begin{cases} 0 & \text{if } c = 0 \\ P(d\mathbf{xy})\log\frac{(\max_{\mathbf{z}\models\mathbf{xy}} P(d\,|\,\mathbf{z}))(1-\min_{\mathbf{x}'\mathbf{y}'} P(\mathbf{xx}'\,|\,\mathbf{yy}'))}{(\min_{\mathbf{z}\models\mathbf{xy}} P(d\,|\,\mathbf{z}))(1-\max_{\mathbf{x}'\mathbf{y}'} P(\mathbf{xx}'\,|\,\mathbf{yy}'))+c} & \text{if } c < 0 \\ P(\bar{d}\mathbf{xy})\log\frac{(\max_{\mathbf{z}\models\mathbf{xy}} P(\bar{d}\,|\,\mathbf{z}))(1-\min_{\mathbf{x}'\mathbf{y}'} P(\mathbf{xx}'\,|\,\mathbf{yy}'))}{(\min_{\mathbf{z}\models\mathbf{xy}} P(\bar{d}\,|\,\mathbf{z}))(1-\max_{\mathbf{x}'\mathbf{y}'} P(\mathbf{xx}'\,|\,\mathbf{yy}'))-c} & \text{if } c > 0 \end{cases}
\end{aligned}$$

E Proof of Proposition 2

The probability values of positive decision in terms of naive Bayes parameters θ are as follows:

$$P_\theta(d\,|\,\mathbf{xy}) = \frac{P_\theta(d\mathbf{xy})}{P_\theta(\mathbf{xy})} = \frac{\theta_d\prod_x\theta_{x\,|\,d}\prod_y\theta_{y\,|\,d}}{\theta_d\prod_x\theta_{x\,|\,d}\prod_y\theta_{y\,|\,d} + \theta_{\bar{d}}\prod_x\theta_{x\,|\,\bar{d}}\prod_y\theta_{y\,|\,\bar{d}}} = \frac{1}{1+\frac{\theta_{\bar{d}}\prod_x\theta_{x\,|\,\bar{d}}\prod_y\theta_{y\,|\,\bar{d}}}{\theta_d\prod_x\theta_{x\,|\,d}\prod_y\theta_{y\,|\,d}}},$$

$$P_\theta(\bar{d}\,|\,\mathbf{y}) = \frac{P_\theta(d\mathbf{y})}{P_\theta(\mathbf{y})} = \frac{1}{1+\frac{\theta_{\bar{d}}\prod_y\theta_{y\,|\,\bar{d}}}{\theta_d\prod_y\theta_{y\,|\,d}}}.$$

For simplicity of notation, let us write:

$$r_\mathbf{x} = \frac{\prod_x\theta_{x\,|\,\bar{d}}}{\prod_x\theta_{x\,|\,d}}, \quad r_\mathbf{y} = \frac{\theta_{\bar{d}}\prod_y\theta_{y\,|\,\bar{d}}}{\theta_d\prod_y\theta_{y\,|\,d}}. \tag{5}$$

Then the degree of discrimination is $\Delta_{P_\theta,d}(\mathbf{x},\mathbf{y}) = P_\theta(d\,|\,\mathbf{xy}) - P_\theta(d\,|\,\mathbf{y}) = \frac{1}{1+r_\mathbf{x}r_\mathbf{y}} - \frac{1}{1+r_\mathbf{y}}$. Now we express the fairness constraint $|\Delta_{P_\theta,d}(\mathbf{x},\mathbf{y})| \leq \delta$ as the following two inequalities:

$$-\delta \leq \frac{(1+r_\mathbf{y}) - (1+r_\mathbf{x}r_\mathbf{y})}{(1+r_\mathbf{x}r_\mathbf{y})\cdot(1+r_\mathbf{y})} \leq \delta.$$

After simplifying,

$$r_\mathbf{y} - r_\mathbf{x}r_\mathbf{y} \geq -\delta(1+r_\mathbf{x}r_\mathbf{y}+r_\mathbf{y}+r_\mathbf{x}r_\mathbf{y}^2), \quad r_\mathbf{y} - r_\mathbf{x}r_\mathbf{y} \leq \delta(1+r_\mathbf{x}r_\mathbf{y}+r_\mathbf{y}+r_\mathbf{x}r_\mathbf{y}^2).$$

We further express this as the following two signomial inequality constraints:

$$\left(\frac{1-\delta}{\delta}\right)r_\mathbf{x}r_\mathbf{y} - \left(\frac{1+\delta}{\delta}\right)r_\mathbf{y} - r_\mathbf{x}r_\mathbf{y}^2 \leq 1, \quad -\left(\frac{1+\delta}{\delta}\right)r_\mathbf{x}r_\mathbf{y} + \left(\frac{1-\delta}{\delta}\right)r_\mathbf{y} - r_\mathbf{x}r_\mathbf{y}^2 \leq 1 \tag{6}$$

Note that $r_\mathbf{x}$ and $r_\mathbf{y}$ according to Equation 5 are monomials of θ, and thus above constraints are also signomial with respect to the optimization variables θ. □

Table 3: 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

Dataset Statistics						Divergence score			Discrimination score		
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

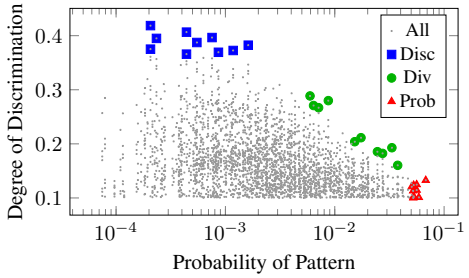


Figure 2: Discrimination patterns with $\delta = 0.1$ for the max-likelihood NB classifier on COMPAS.

F Additional Experiments

Table 3 shows a summary of the three datasets used and the fraction of patterns explored by our discrimination pattern miner, on different rank heuristics and k and δ values.

To study the efficacy of the divergence score in finding interesting patterns, we plot all discrimination patterns in the COMPAS dataset (see Figure 2). The top-10 patterns according to three measures (discrimination, divergence, and probability) are highlighted. The observed trade-off between probability and discrimination score indicates that the top patterns according to one measure are ranked low according to the other measure. The divergence score, however, balances the two measures and returns patterns that have high probability and discrimination scores. Moreover, the patterns selected by the divergence score lie on the Pareto front of probability and discrimination score. This in fact always holds by definition; fixing the probability and increasing the discrimination score also increases the divergence score, and vice versa.

Figure 3 shows the number of iterations it takes for our algorithm to learn a δ-fair model, on the three datasets with varying values of k and δ. As noted earlier, all instances converge in a small number of iterations.

Table 4: Number of remaining patterns with $\delta = 0.1$ in naive Bayes models trained on discrimination-free data.

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

Lastly, we empirically demonstrate that discrimination patterns still occur when learning naive Bayes models from fair data. We use the data repair algorithm proposed by Feldman et al. [6] to remove discrimination from data, and learn a naive Bayes model from the repared data. Table 4 shows the number of remaining discrimination patterns in such model, where λ determines the tradeoff between fairness and accuracy in the data repair step. 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.

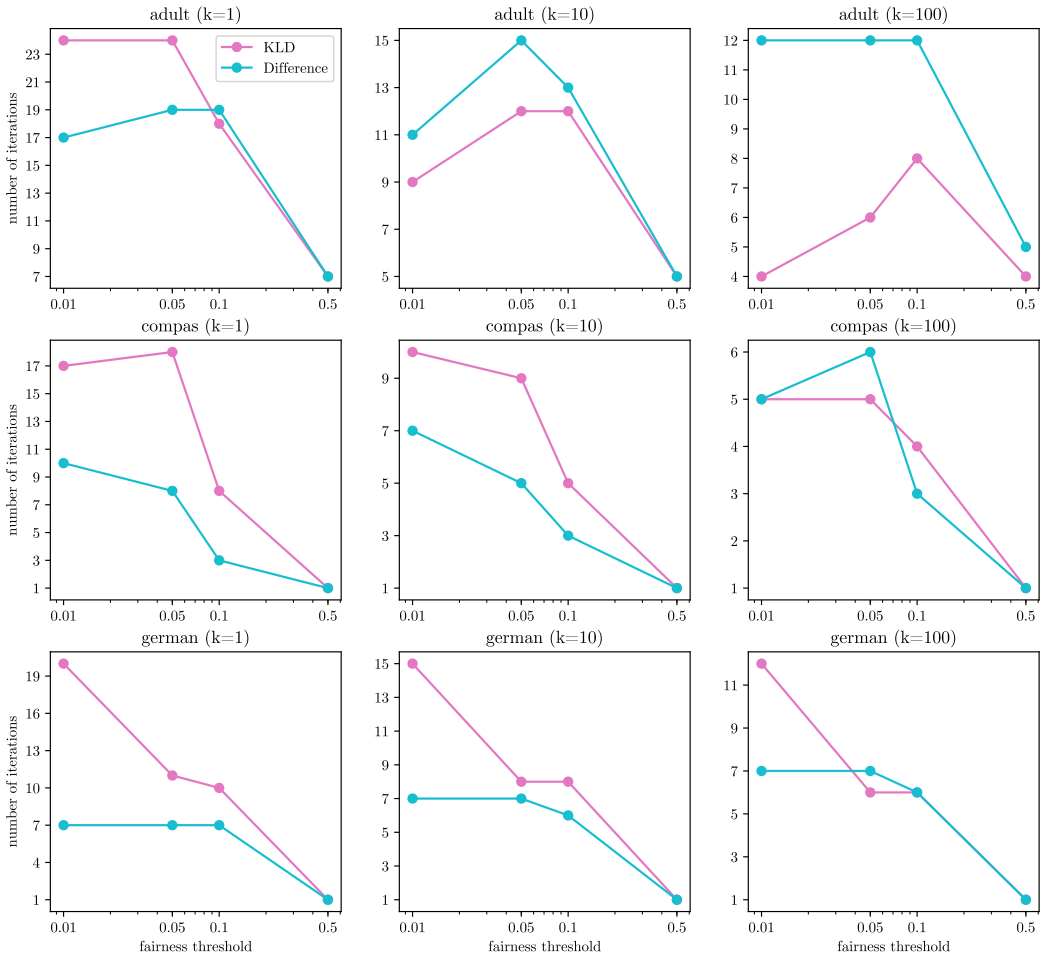


Figure 3: Number of iterations of δ-fair learner until convergence