Learning Fair Naive Bayes Classifiers by Discovering and Eliminating Discrimination Patterns

by (Your Name)

Roadmap

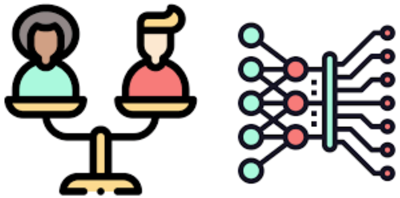

1
Introduction
2
Problem Formalization
3
Searching for Discrimination Patterns
4
Learning Fair Naive Bayes Classifiers
5
Discussion and Conclusion

Introduction

Fairness in machine learning refers to the various attempts at correcting algorithmic bias in automated decision processes based on machine learning models.

Examples:

- Racial and gender bias in image recognition algorithms.
- 'COMPAS' software, widely used in US courts to predict recidivism
- Automatic tagging feature in both Flicker and Google Photos



Contributions

- Discrimination Pattern refers to an individual receiving different classifications depending on whether some sensitive attributes were observed.
- 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.

Formalizing Problem

- *Let* P *be a distribution over* D ∪ Z.

  Let x *and* y *be joint assignments to* X ⊆ S *and* Y ⊆ Z \ X.

  The degree of discrimination *of* xy *is:*

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

Formalizing Problem

- *Let* P *be a distribution over* D ∪ Z, *and* $\delta \in [0, 1]$ *a threshold.*

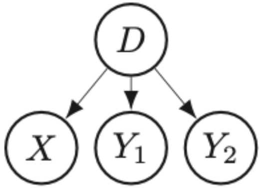
Joint assignments x *and* y *form a* discrimination pattern *w.r.t.* P *and* δ *if:*

(1) $X \subseteq S$ *and* $Y \subseteq Z \setminus X$;

and

(2) $|\Delta_{P,d}(x, y)| > \delta$.

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



$P(d)$
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

The network is individually fair for $\delta = 0.2$ because $\max_{xy1y2} |\Delta(x, y1y2)| = 0.167 \leq \delta$.

However…

👉 $\Delta(\bar{x},y1)| = 0.225 > \delta$

Big Challenge

Computing the degree of discrimination involves probabilistic inference, which is hard in general, and a given distribution may have exponentially many patterns…

Searching for Discrimination Patterns

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: $\mathbf{x} \leftarrow \emptyset$, $\mathbf{y} \leftarrow \emptyset$, $\mathbf{E} \leftarrow \emptyset$, $L \leftarrow []$

---

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

Top-k Patterns

- Nevertheless, ranking patterns by their discrimination score may return patterns of very low probability.
- patterns with higher divergence score will tend to have not only higher discrimination score but also higher probabilities.

Kullback-Leibler divergence

- *Let* P *be a distribution over* D ∪ Z. *Let* x *and* y *be joint instantiations to subsets* X ⊆ S *and* Y ⊆ Z \ X.

  The divergence score *of* xy *is:*

$$\mathrm{Div}_{P,d,\delta}(\mathbf{x},\mathbf{y}) \triangleq \min_{Q} \mathrm{KL}\left(P \parallel Q\right)$$

$$\textit{s.t.} \quad |\Delta_{Q,d}(\mathbf{x},\mathbf{y})| \leq \delta$$

$$P(d\mathbf{z}) = Q(d\mathbf{z}),\ \forall\, d\mathbf{z} \not\models \mathbf{xy}$$

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

Empirical Evaluation of Discrimination Pattern Miner

- 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.
- We use three datasets:

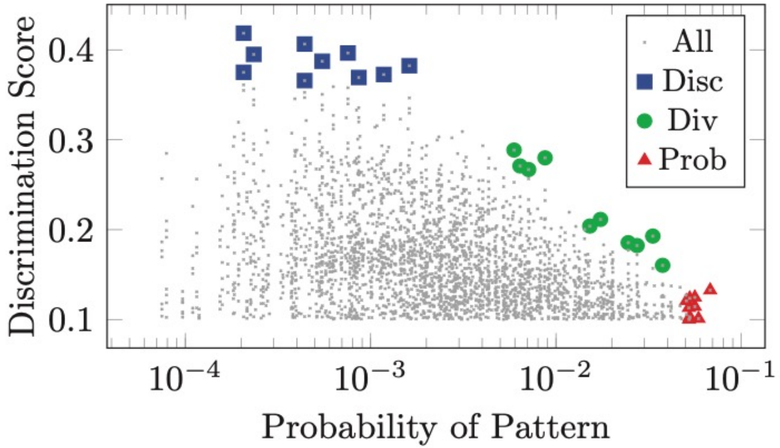
The *Adult* dataset and *German* dataset are used for predicting income level and credit risk

the *COMPAS* dataset is used for predicting recidivism.

Q1: Does our pattern miner find discrimination patterns more efficiently than by enumerating all possible patterns?

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

Q2: Does the divergence score find discrimination patterns with both a high discrimination score and high probability?



Learning Fair Naive Bayes Classifiers

- We formulate the learning subject to fairness constraints as a signomial program, which has the following form:

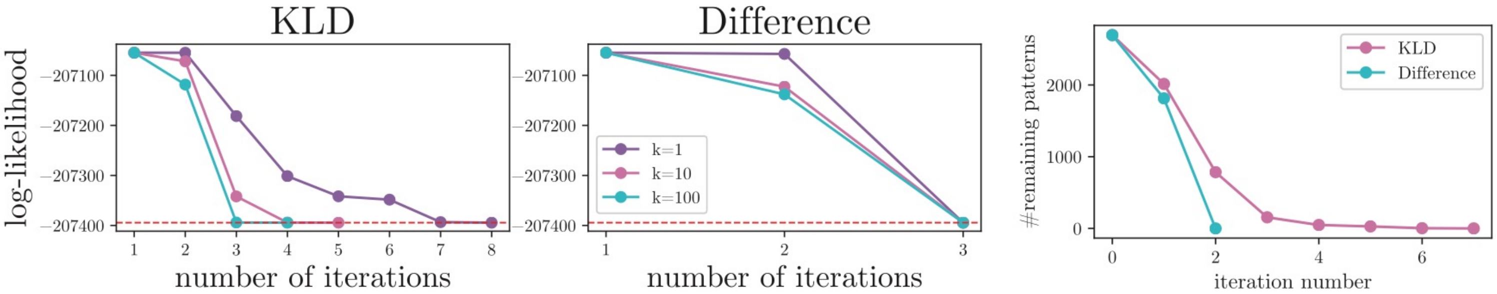
$$\text{minimize } f_0(x), \quad \text{s.t.} \quad f_i(x) \leq 1, \quad g_j(x) = 1 \quad \forall\, i, j$$

- f_i is signomial while g_j is monomial. A *signomial* is a function of the form $\sum_k c_k x_1^{a_{1k}} \cdots x_n^{a_{1n}}$ where c_k, $a_{ij} \in R$;

a *monomial* is of the form $c x_1^{a_1} \cdots x_n^{a_n}$ where $c>0$, $a_{ij} \in R$;

- Signomial programs are not globally convex, but a locally optimal solution can be computed efficiently.

Q1. Can we learn a δ-fair model in a small number of iterations while only asserting a small number of fairness constraints?



Q2. How does the performance of δ-fair naive Bayes classifier compare to existing work?

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

Conclusion

- we introduced a novel definition of fair probability distribution in terms of discrimination patterns
- presented algorithms to search for discrimination patterns in naive Bayes networks and to learn a high quality fair naive Bayes classifier from data.
- Our algorithm is only a tool to assist such experts in learning fair distributions

Thanks!

Any questions?

You can find me at:

(Your mail id)@yahoo.com