Fair and Interpretable Decision Rules for Binary Classification

Connor Lawless[*] , **Oktay Günlük** ,
Cornell University
{cal379, ong5}@cornell.edu

Abstract

In this paper we consider the problem of building Boolean rule sets in disjunctive normal form (DNF), an interpretable model for binary classification, subject to fairness constraints. We formulate the problem as an integer program that maximizes classification accuracy with explicit constraints on two different measures of classification parity: equality of opportunity, and equalized odds. A column generation framework, with a novel formulation, is used to efficiently search over exponentially many possible rules, eliminating the need for heuristic rule mining. Compared to CART and Logistic Regression, two interpretable machine learning algorithms, our method produces interpretable classifiers that have superior performance with respect to both fairness metrics.

1 Introduction

With the explosion of artificial intelligence in recent years, automated decision making has begun taking over key decision making tasks in a variety of areas ranging from finance to driving. However, with machine learning dictating decisions as important as lending, hiring, and college admissions, a natural question is whether these algorithms are fair to all those affected. Recent results have shown machine learning algorithms to be racially biased in a range of applications ranging from facial identification in picture tagging to predicting criminal recidivism [24]. Further complicating the problem is the need for model interpretability in many applications where machine learning models complement human decision making, such as criminal justice and medicine. In these applications transparency is necessary for domain experts to understand, critique and trust machine learning models. With these dueling objectives in mind, practitioners face the daunting question of whether it is possible to design machine learning algorithms that are accurate, fair AND interpretable. This paper takes one step towards such an algorithm for supervised binary classification problems using integer programming to build interpretable rule sets that can explicitly include fairness constraints.

[*]Contact Author

We focus on a well-studied interpretable class of rule sets, Boolean rules in disjunctive normal form (DNF, 'OR-of-ANDs'). For example, a DNF rule set with two rules for predicting criminal recidivism could be

$$\big[\text{(Priors}\geq 3\text{) and (Age}\leq 45\text{) and (Score Factor} = \text{TRUE)}\big]$$
$$\text{OR}$$
$$\big[\text{(Priors}\geq 20\text{) and (Age}\geq 45\text{)}\big]$$

where Priors, Age, and Score Factor are features related to the defendant. The fewer the rules or conditions in each rule, the more interpretable the rule set. In contrast to decision trees [7; 25; 6; 2; 18], and decision lists [26; 23; 27; 3; 22; 28], other interpretable classes of rule sets, the rules within a DNF rule-set are unordered and have been shown in a user study to require less effort to understand [21]. Practically, optimal decision rules have been shown to be more accurate than heuristic rule set methods [11], while remaining more computationally tractable than other optimal rule set methods [15; 6]. To build fair DNF rule sets, we start with the model in [11] which frames the problem as a large integer program (IP), generating candidate rules using a column generation (CG) framework. We keep the objective of the IP the same and add explicit constraints on fairness to control the level of acceptable "unfairness" among different subgroups. We also add additional constraints to the model as the objective function does not guarantee correctness in the presence of fairness constraints. Our approach also differs from [11] in the way we solve the pricing problem as we use a very compact formulation to generate candidate rules which reduces the computational effort significantly.

2 Fairness Metrics

We start by defining the standard supervised binary classification problem, where given a training set of n samples $(\mathbf{X}_i, y_i)$ with labels $y_i \in \{0,1\}$ and features $X_i \in \{0,1\}^p$ for $i \in I = \{1,\ldots,n\}$, the goal is to generate a decision rule $d : \{0,1\}^p \rightarrow \{0,1\}$ that minimizes the expected error $\mathbb{P}(d(X) \neq Y)$ between the predicted label and the true label for unseen data. Assuming the data to be binary-valued, as seen in [11; 15], is not restrictive as numeric features can be *binarized* using a sequence of thresholds and the same can be done for categorical features using one-hot encoding.

Now consider the case when each data-point also has an associated group (or protected feature) $g_i \in \mathcal{G}$ where $\mathcal{G}$ is a given discrete set. Quantifying fairness is not a straight forward task in this context and a number of metrics have been proposed in the fair machine learning literature. One popular category of fairness metrics is classification party [13; 1; 8; 14; 19; 17; 1; 29; 17] - which ensures that some measure of prediction error (ex. Type I/II error, accuracy) is equal across all groups. We focus on two measures of classification party: Equality of Opportunity and Equalized Odds. Both criteria fit naturally into the integer programming formulation presented in Section 3 and have a number of real world applications such as credit lending and hiring.

Equality of Opportunity: This fairness criterion requires the false negative rate to be equal across groups by enforcing the following condition [17]:

$$\mathbb{P}(d(X)=0|Y=1,G=g)=\mathbb{P}(d(X)=0|Y=1) \quad (1)$$

for all $g \in \mathcal{G}$. This condition is particularly relevant when there is a much larger societal cost to false negatives than false positives, for example in applications such as loan approval or hiring decisions, see [10; 30].

Equalized odds: A stricter condition on the classifier is to require that the classification error is equal across all groups and for both the positive and negative classes within those groups [17]. To achieve equalized odds, together with equation (1), the following condition is also enforced:

$$\mathbb{P}(d(X)=1|Y=0,G=g)=\mathbb{P}(d(X)=1|Y=0) \quad (2)$$

for all $g \in \mathcal{G}$.

In a practical setting, it is unrealistic to expect to find classifiers that can satisfy the above criteria exactly and therefore one needs to consider how much these conditions are violated as a measure of fairness. For example, in the context of equality of opportunity, the maximum disparity among groups can be used to measure the *unfairness* of a given classifier d as follows:

$$\Delta(d)=\max_{g,g'\in\mathcal{G}}\Big|\mathbb{P}(d(X)=0|Y=1,G=g)- \mathbb{P}(d(X)=0|Y=1,G=g')\Big|.$$

When training the classifier d, one can then use $\Delta(d)$ in the objective function as a penalty term or can explicitly require a constraint of the form $\Delta(d)\le\epsilon$ to be satisfied by d. We focus on the latter case as it allows for explicit control over tolerable unfairness. A more precise discussion of how this constraint is integrated is included in Section 3.

3 Classification Framework: Boolean Decision Rule Sets

We now introduce our method to construct optimal DNF rule-sets for binary classification subject to fairness constraints. Note that when the input data is binary-valued, a DNF-rule set simply corresponds to checking whether a subset of features satisfies a specific combination of 0s and 1s. By ensuring that the data also includes the complement of every feature, a DNF-rule set simply checks if a subset of features are all 1 for a given data point. Consequently, if there are p binary features there can only be a finite number of (2^p-2) possible decision rules. Therefore, in theory, it is possible to enumerate all possible rules and then formulate a large integer program (IP) to select a small subset of them to minimize error on the training data under explicit fairness constraints. However from a practical perspective, it is clearly not possible to solve an exponential size IP, so instead we solve the continuous relaxation (LP) of the IP using column generation. Consequently, instead of enumerating all possible rules, we only enumerate those that can potentially improve classification error. Similar to [11] the objective of the IP is to minimize *Hamming loss*, a proxy for classification error that counts the number of rules that needs to be changed to classify a point correctly. From a practical perspective, Hamming loss leads to smaller IP formulations that can be solved more efficiently.

3.1 Integer Program Formulation

Let $\mathcal{K}$ denote the set of all possible DNF rules and $\mathcal{K}_i\subset\mathcal{K}$ be the set of rules met by data point $i\in I$. Let c_k denote the complexity of rule $k\in\mathcal{K}$ which is defined as a fixed cost of 1 plus the number of conditions in the rule. Assume that the data points are partitioned into two sets based on their labels: $\mathcal{P}=\{i\in I: y_i=1\}$, and $\mathcal{Z}=\{i\in I: y_i=0\}$. Additionally, for each group $g\in\mathcal{G}$ we denote the data points that have the protected feature g with $\mathcal{G}_g=\{i\in I: g_i=g\}$ and let $\mathcal{P}_g=\mathcal{P}\cap\mathcal{G}_g$ and $\mathcal{Z}_g=\mathcal{Z}\cap\mathcal{G}_g$. For simplicity, we describe the constraints assuming $\mathcal{G}=\{1,2\}$ and note that extending it to multiple groups is straightforward. Let $w_k\in\{0,1\}$ be a variable indicating if rule $k\in\mathcal{K}$ is selected; $\zeta_i\in\{0,1\}$ be a variable indicating if data point $i\in\mathcal{P}$ is misclassified; and C be a parameter denoting the maximum complexity allowed. With this notation in mind, the problem of identifying the optimal rule set subject to constraints on *equality of opportunity* becomes:

$$z_{mip}=\mathbf{min}\sum_{i\in\mathcal{P}}\zeta_i+\sum_{i\in\mathcal{Z}}\sum_{k\in\mathcal{K}_i}w_k \quad (3)$$

$$\textbf{s.t.}\quad \zeta_i+\sum_{k\in\mathcal{K}_i}w_k\ge 1 \quad i\in\mathcal{P} \quad (4)$$

$$C\zeta_i+\sum_{k\in\mathcal{K}_i}2w_k\le C \quad i\in\mathcal{P} \quad (5)$$

$$\sum_{k\in\mathcal{K}}c_kw_k\le C \quad (6)$$

$$w\in\{0,1\}^{|\mathcal{K}|},\ \zeta\in\{0,1\}^{|\mathcal{P}|} \quad (7)$$

$$\frac{1}{|\mathcal{P}_1|}\sum_{i\in\mathcal{P}_1}\zeta_i-\frac{1}{|\mathcal{P}_2|}\sum_{i\in\mathcal{P}_2}\zeta_i\le\epsilon_1 \quad (8)$$

$$\frac{1}{|\mathcal{P}_2|}\sum_{i\in\mathcal{P}_2}\zeta_i-\frac{1}{|\mathcal{P}_1|}\sum_{i\in\mathcal{P}_1}\zeta_i\le\epsilon_1 \quad (9)$$

We denote this integer program the Master Integer Program (MIP), and it's associated linear relaxation the Master LP (MLP) (obtained by dropping the integrality constraint). Any feasible solution $(\bar{w},\bar{\zeta})$ to (4)-(7) corresponds to a rule set $S=\{k\in\mathcal{K}: \bar{w}_k=1\}$. Note that the objective is the Hamming loss where the first term counts the number of misclassified data-points ζ_i for $i\in\mathcal{P}$, whereas the second term adds up the total number of selected rules satisfied by data-points i for each $i \in \mathcal{Z}$. Constraint (4) identifies false negatives by forcing ζ_i to take value 1 if no rule that is satisfied by the point $i \in \mathcal{P}$ is selected. Within the constraint we multiply w_k by 2 as it is the minimum complexity for a rule. Similarly, constraint (5) ensures that ζ_i can only take a value of 1 if no rules satisfied by $i \in \mathcal{P}$ are selected. Here we use the fact that $c_k \geq 2$ for all $k \in \mathcal{K}$. Constraint (6) provides the bound on complexity of the final rule set. Finally, constraints (8) and (9) bound the maximum allowed unfairness, denoted by Δ in section 2 by a specified constant $\epsilon_1 \geq 0$. If ϵ_1 is chosen to be 0, then the fairness constraint is imposed strictly. Depending on the application, ϵ_1 can also be larger than 0, in which case a prescribed level of unfairness is tolerated.

Hamming Equalized Odds (HEO): We next extend the notion of equalized odds to the hamming loss setting (henceforth denoted hamming equalized odds). Specifically, to bound the disparity in false positive rate we bound the disparity in the hamming loss terms for the negative class. To that end, together with constraints (8) and (9) , we include the following constraints in the formulation:

$$\frac{1}{|\mathcal{Z}_1|}\sum_{i\in\mathcal{Z}_1}\sum_{k\in\mathcal{K}_i} w_k - \frac{1}{|\mathcal{Z}_2|}\sum_{i\in\mathcal{Z}_2}\sum_{k\in\mathcal{K}_i} w_k \leq \epsilon_2 \tag{10}$$

$$\frac{1}{|\mathcal{Z}_2|}\sum_{i\in\mathcal{Z}_2}\sum_{k\in\mathcal{K}_i} w_k - \frac{1}{|\mathcal{Z}_1|}\sum_{i\in\mathcal{Z}_1}\sum_{k\in\mathcal{K}_i} w_k \leq \epsilon_2, \tag{11}$$

where $\epsilon_2 \geq 0$ is a given constant. The tolerance parameter ϵ_2 can be set equal to ϵ_1 or it can be chosen separately. Note that we normalize the hamming loss terms to account for the difference in group sizes and positive response rates between groups.

3.2 Column Generation Framework

To solve the LP relaxation of the MIP, called the MLP, using the column generation framework [9], we start with a small subset $\hat{\mathcal{K}} \subset \mathcal{K}$ of all possible rules and solve an LP restricted to the variables associated with these rules only. Once this small LP is solved, we use its optimal dual solution to identify a missing variable (rule) that has a negative reduced cost [5]. The search for such a variable is called the *pricing problem* and, in our case, this can be done by solving a separate integer program. If a variable with a negative reduced cost is found, then $\hat{\mathcal{K}}$ is augmented with the associated rule and the LP is solved again and the this process is repeated until no such variables can be found.

Given a possibly empty subset of rules $\hat{\mathcal{K}} \subset \mathcal{K}$, let the restricted MLP, defined by (3)-(6), (8)-(9) and denoted by RMLP, be the restriction of MLP to the rules in $\hat{\mathcal{K}}$. Let $(\mu, \alpha, \lambda, \gamma^1, \gamma^2)$ be an optimal *dual* solution to RMLP, where variables $\mu, \alpha, \lambda \geq 0$ are associated with constraints (4), (5), and (6), respectively. Variables γ^1 and γ^2 are associated with fairness constraints (8) and (9). We now formulate an integer program to find a $k \in \mathcal{K}$ with the minimum reduced cost $\hat{\rho}_k$. Remember that a decision rule corresponds to a subset of the binary features J and classifies a data point with a positive response if the point has all the features selected by the rule. Let variable $z_j \in \{0, 1\}$ for $j \in J$ denote if the rule has feature j and let variable $\delta_i \in \{0, 1\}$ for $i \in I$ denote if the rule misclassifies sample i. Using these variables, the complexity of a rule can be computed as $(1 + \sum_{j\in J} z_j)$. We now construct the full pricing problem with the reduced cost in the objective:

$$z_{cg} = \ \mathbf{min} \sum_{i\in\mathcal{Z}} \delta_i + \sum_{i\in\mathcal{P}}(2\alpha_i - \mu_i)\delta_i + \lambda(1 + \sum_{j\in J} z_j)$$

$$\mathbf{s.t.} \quad D\delta_i + \sum_{j\in S_i} z_j \leq D \quad i \in I^- \tag{12}$$

$$\delta_i + \sum_{j\in S_i} z_j \geq 1 \quad i \in I^+ \tag{13}$$

$$\sum_{j\in J} z_j \leq D \tag{14}$$

$$z \in \{0, 1\}^{|J|},\ \delta \in \{0, 1\}^{|\mathcal{P}|} \tag{15}$$

where the set $I^- \subseteq I$ contains the indices of δ_i variables that have a negative coefficient in the objective, and $I^+ = I \setminus I^-$. The objective is the reduced cost for the generated rule. Note that variable w_k does not appear in constraints (8) or (9) in RMLP and consequently the objective does not involve variables γ^1 or γ^2. Also note that constraints (12) and (13) to ensure that δ_i accurately reflects whether the new rule classifies data point i with a positive label, and constraint (14) puts an explicit bound on the complexity of any rule using the parameter D. This individual rule complexity constraint can be set independently of C in the master problem or simply be set to $C - 1$.

Hamming Equalized Odds: In this case the RMLP is defined by (3)-(6), (8)-(11) and note that unlike (8) and (9), constraints (10) and (11) do involve variables w_k. Let $(\mu, \alpha, \lambda, \gamma^1, \gamma^2, \gamma^3, \gamma^4)$ be an optimal dual solution to RMLP, where variables γ^3 and γ^4 are associated with fairness constraints (10) and (11), respectively. Using this dual solution, we update the objective to be:

$$z_{cg} \ = \ \mathbf{min}\ (1 + \frac{\gamma_3 - \gamma_4}{|\mathcal{Z}_1|}) \sum_{i\in\mathcal{Z}_1} \delta_i + (1 + \frac{\gamma_4 - \gamma_3}{|\mathcal{Z}_2|}) \sum_{i\in\mathcal{Z}_2} \delta_i + \sum_{i\in\mathcal{P}}(2\alpha_i - \mu_i)\delta_i + \lambda(1 + \sum_{j\in J} z_j)$$

4 Experiments

We implemented the fair column generation algorithm (denoted FairCG) using the Python interface of Gurobi [16] to solve the linear and integer programs. To solve the MLP we use a barrier interior point method with the default crossover parameter. For the pricing problem we use the default settings, and return all solutions generated during the algorithm's run with negative reduced costs. For each of the experiments we set a time limit of 5 minutes for the overall training, and a limit of 45 seconds to solve the pricing problem.

To benchmark the performance of our algorithm, we tested it on three fair machine learning datasets: *default* [12], *adult* [12], and *compas* [4]. Figure 1 (a) shows the trade-off between the fairness constraint for equality of opportunity when training and the realized false negative rate. As we relax the

Table 1: Mean Accuracy and Fairness Results for Equality of Opportunity

		Adult		Compas		Default	
		Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
Fair CG	Best Acc	82.9 (0.2)	9.4 (0.4)	68.2 (1.2)	24.5 (5.3)	82.0 (0.6)	0.5 (1.2)
	Best Fair	78.4 (0.4)	0.3 (0.3)	53.0 (1.6)	0 (0)	77.9 (0.4)	0 (0)
CART	Best Acc	85.5 (0.3)	15.9 (5.1)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Best Fair	85.4 (0.5)	8.4 (4.3)	65.8 (2.3)	21.3 (6.1)	82.0 (1.4)	2.5 (1.9)
LR	Best Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Best Fair	79.8 (0.6)	3.6 (3.2)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)

Table 2: Mean Accuracy and Fairness Results for Hamming Equalized Odds

Fair CG	Tuned for Acc	83.1 (0.6)	7.9 (0.4)	67.5 (1.7)	24.5 (5.3)	81.9 (0.6)	0.9 (1.1)
	Tuned for Fair	76.0 (0.5)	0 (0)	53.0 (1.7)	0 (0)	81.9 (0.6)	0 (0)
CART	Tuned for Acc	85.5 (0.3)	7.2 (0.5)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Tuned for Fair	85.3 (0.5)	6.8 (0.5)	66.8 (2.6)	16.8 (5.4)	82.0 (1.3)	1.1 (0.6)
LR	Tuned for Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Tuned for Fair	79.8 (0.6)	1.0 (0.5)	67.5 (1.2)	19.1 (4.5)	77.9 (1.7)	0 (0)

fairness constraint both the realized train and test set fairness decreases (i.e. the gap between groups grows). This disproportionately impacts the false negative rate for the first group, underscoring the importance of finding fair classifiers. Figure 1 (b) shows that increasing rule set complexity leads to lower false negative rates across groups, underscoring the inherent trade-off in interpretability and fairness as discussed in [20]. Results for HEO and other data-sets show similar trends.

We also compared the performance of our algorithm with two other interpretable binary classification models, CART and Logistic Regression. For all three models we varied the hyper-parameters, performing 10-fold cross validation for each parameter, to generate the accuracy fairness trade-offs. These models used the real-valued feature data. Figure 1 (c) plots the efficient frontier for accuracy-fairness for CART, Logistic Regression as well as our own algorithm for the *compas* dataset. Tables 1 and 2 summarize each algorithm's performance when tuned for accuracy and fairness separately. For each dataset we report the standard deviation in parenthesis. While CART is able to achieve superior predictive accuracy on some datasets, our algorithm is able to achieve comparable accuracy under much stricter fairness constraints. This shows that our framework is able to build interpretable models that have competitive accuracy and substantially improved fairness.

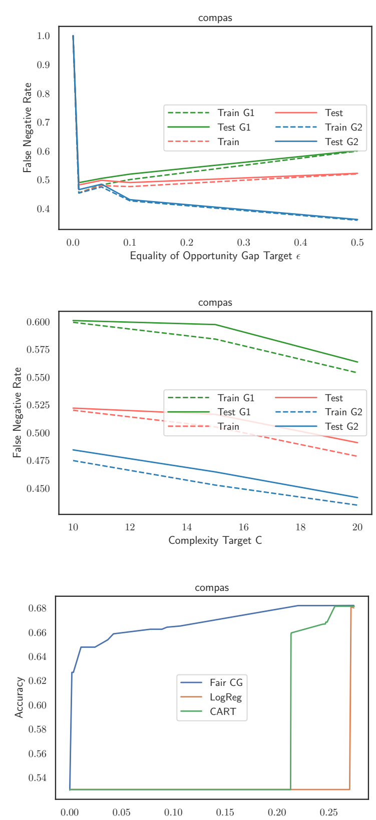

Figure 1: Impact of the equality of opportunity fairness constraint (a) and complexity constraint (b) on false negative rate for the *compas* dataset. (c) Performance of FairCG benchmarked against other interpretable models on *compas* dataset. For all plots, if group is unspecified line is for all data.

5 Conclusion

While many practitioners have explored the problem of building fair or interpretable classification models, few have looked at the increasingly important problem of creating fair *and* interpretable models. In this work we begin bridging that gap, using an integer programming approach. Preliminary empirical results show that our algorithm can achieve competitive accuracy on standard fair ML datasets with superior fairness when compared against simple interpretable models.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification, 2018.

[2] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making, 2019.

[3] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 35–44, 2017.

[4] Julia Angwin, Jeff Larson, Surya Mattu, and Laura Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. Available at https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (2016).

[5] Mokhtar S. Bazaraa, John Jarvis, and Hanif D. Sherali. *Linear programming and network flows*. Wiley, 2010.

[6] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Mach. Learn.*, 106(7):1039–1082, July 2017.

[7] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[8] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, 21:277–292, 09 2010.

[9] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. *Integer programming*. Springer, 2014.

[10] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning, 2018.

[11] Sanjeeb Dash, Oktay Günlük, and Dennis Wei. Boolean decision rules via column generation, 2018.

[12] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness through awareness, 2011.

[14] Harrison Edwards and Amos Storkey. Censoring representations with an adversary, 2015.

[15] Oktay Gunluk, Jayant Kalagnanam, Minhan Li, Matt Menickelly, and Katya Scheinberg. Optimal generalized decision trees via integer programming, 2019.

[16] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020.

[17] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning, 2016.

[18] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, page 869–874, USA, 2010. IEEE Computer Society.

[19] Faisal Kamiran, Indrė Žliobaitė, and Toon Calders. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowledge and Information Systems*, 1:in press, 06 2012.

[20] Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: Implications for fairness, stereotypes, and interpretability, 2019.

[21] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1675–1684, New York, NY, USA, 2016. Association for Computing Machinery.

[22] Himabindu Lakkaraju and Cynthia Rudin. Learning cost-effective and interpretable treatment regimes. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 166–175, Fort Lauderdale, FL, USA, 20–22 Apr 2017.

[23] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.*, September(3):1350–1371, 09 2015.

[24] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning, 2019.

[25] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[26] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[27] Tong Wang and Cynthia Rudin. Learning Optimized Or's of And's, November 2015. arXiv:1511.02210.

[28] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable Bayesian rule lists. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1013–1022, 2017.

[29] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact. *Proceedings of the 26th International Conference on World Wide Web - WWW '17*, 2017.

[30] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification, 2015.