Learning Fair Rule Lists

Ulrich Aïvodji, Julien Ferry, Sébastien Gambs, Marie-José Huguet, Mohamed Siala

Learning Fair Rule Lists

Ulrich Aïvodji
aivodji.ulrich@courrier.uqam.ca
Université du Québec à Montréal
Montreal, Canada

Julien Ferry*
jferry@laas.fr
LAAS-CNRS
Toulouse, France

Sébastien Gambs
gambs.sebastien@uqam.ca
Université du Québec à Montréal
Montreal, Canada

Marie-José Huguet
huguet@laas.fr
LAAS-CNRS, Université de Toulouse, CNRS, INSA
Toulouse, France

Mohamed Siala
msiala@laas.fr
LAAS-CNRS, Université de Toulouse, CNRS, INSA
Toulouse, France

ABSTRACT

The widespread use of machine learning models, especially within the context of decision-making systems impacting individuals, raises many ethical issues with respect to fairness and interpretability of these models. While the research in these domains is booming, very few works have addressed these two issues simultaneously. To solve this shortcoming, we propose FairCORELS, a supervised learning algorithm whose objective is to learn at the same time fair and interpretable models. FairCORELS is a multi-objective variant of CORELS, a branch-and-bound algorithm, designed to compute accurate and interpretable rule lists. By jointly addressing fairness and interpretability, FairCORELS can achieve better fairness/accuracy tradeoffs compared to existing methods, as demonstrated by the empirical evaluation performed on real datasets. Our paper also contains additional contributions regarding the search strategies for optimizing the multi-objective function integrating both fairness, accuracy and interpretability.

1 INTRODUCTION

Machine learning models are now becoming more and more common in high stake decision-making systems (*e.g.*, credit scoring [56], predictive justice [37] and automatic recruiting [47]). These systems can have an important impact on individuals as decisions based on wrong predictions can adversely affect human lives (*e.g.*, people being wrongly denied parole [60]). Thus, ethical aspects such as the fairness and transparency of machine learning models have become not only a desirable feature but also legal requirements. For instance, the European General Data Protection Regulation (GDPR) has a provision requiring explanations of the rationale for the decisions of machine learning models that have a significant impact on individuals [26]. In fact, understanding these models can be considered as a prerequisite towards quantitatively evaluating other criteria such as fairness, reliability and robustness [7, 20].

Two main approaches have emerged in the literature to facilitate the understanding of machine learning models: *black-box explanation* and *transparent-box design* [27, 42, 44, 48]. Black-box explanations techniques consist in providing *a posteriori* explanations that are either global [17, 41] or local approximations [53] of the black-box model. In contrast, transparent-box design aim at building transparent models, which are inherently interpretable [4, 9, 43, 57]. For instance, when they have small or of reasonable size [44], the following models can be considered as interpretable: rules sets [18, 43, 46, 54], rule lists [3, 59, 61], decision trees [9, 51] and scoring systems [39, 57, 64]. For the rest of the paper, we focus on rule lists, which are the interpretable model that we have considered in our work.

With respect to fairness, a significant amount of work has been done in recent years to design fairness-aware machine learning models [12, 25], which we will review in Section 2.2. Nonetheless, despite the progress made in both directions, very few works have focused on learning interpretable models that are simultaneously fair and accurate.

To address this issue, in this paper, we propose FairCORELS: a supervised learning algorithm whose objective is to learn at the same time fair and interpretable models. FairCORELS leverages on the recent advancements, provided by CORELS, for learning certifiably optimal rule lists [3, 4] by adapting them to also take into account fairness constraints. In particular, given a statistical notion of fairness and a sensitive attribute, our algorithm searches for the rule list optimizing the decrease of both unfairness and classification error.

More precisely, the main contributions of our work can be summarized as follows.

- We formulate the problem of learning fair rule lists as a multi-objective version of the problem addressed by CORELS [3, 4]. Then, we propose FairCORELS, a supervised learning algorithm which aims at designing fair rule lists with higher accuracy.
- We introduce a bi-objective method to face both fairness and accuracy. This method benefits from guarantees provided by CORELS on the quality of the solution.
- We design a new lower bound for the unfairness objective to prune the search tree expanded by CORELS. In additions, we devise several branching strategies to improve the effectiveness of the tree search expansion for fair rule lists.
- We evaluate FairCORELS on four public datasets, using statistical parity as the fairness measure, and compare its performances to previous results.

The outline of the paper is as follows. First in Section 2, we review the background notions on fairness and multi-objective optimization necessary to the understanding of our work. Afterwards in Section 3, we introduce our multi-objective optimization framework for learning fair rule lists before describing FairCORELS, our

*Principal contributor.

learning algorithm for realizing this task. Finally, we report on the experiments conducted in Section 4 before concluding with research directions for future works in Section 5.

2 PRELIMINARIES

In this section, we review the notions necessary to the understanding of our work by first examining the relevant literature on fairness metrics and learning methods for fairness enhancement. Afterwards, we introduce rule lists as well as CORELS, a supervised learning algorithm for producing certifiably optimal rule lists. Finally, we give some details on multi-objective optimization.

2.1 Fairness metrics

The literature on fairness in machine learning has boomed in recent years, and it would be impossible to provide a complete review of the existing fairness notions in this paper. We refer the interested reader to the following surveys [6, 14, 50, 58] for a detailed overview of these notions. In a nutshell, three families of fairness notions have emerged in recent years: *causal notions of fairness* [36, 40, 49], which rely on causal assumptions to estimate the effects of sensitive attributes and build algorithms that ensure a tolerable level of discrimination with respect to these attributes, *individual notions of fairness* [21, 29], which require the same decisions for individuals that are similar, and *statistical notions of fairness*.

In this work, we focus primarily on statistical definitions of fairness in our multi-objective optimization framework. This family of fairness notions requires the approximate parity of some statistical measure f across a given set of subgroups of the population according to a particular sensitive attribute that can lead to discrimination (*e.g.*, female or male). For simplicity, we focus on the case of two subgroups hereafter referred to as respectively the minority group and the majority group. In general, the statistical measure f is a function of the predictions of the machine learning model on the different subgroups. For instance in this paper, we have used statistical parity [10, 21, 23, 33, 66] as fairness notion. Other commonly used definitions include predictive parity [13, 38], predictive equality [13, 16], equal opportunity [28] equalized odds [13, 28, 38, 62] and conditional use accuracy equality [6]. A description of these statistical measures can be found in Table 1.

Let $X = \cup_{i=1}^{N} x_i$ be a set of data points where each data point $x_i \in \{0,1\}^J$ is a list of J binary attributes. We assume that X has a 2-partition $P = \{P_{min}, P_{maj}\}$ where P_{min} is called the minority group and P_{max} is called the majority group. Let $Y = \cup_{i=1}^{N} y_i$ where y_i corresponds to a decision attribute (value) for x_i. We define a dataset $\mathcal{D}$ as a triplet (X, P, Y). We assume that the set of attributes is divided into protected (sensitive) and unprotected attributes. The sensitive attributes are used to defined the minority and majority groups. The notion of *unfairness* with respect to a particular classifier and dataset can be defined as followed.

Definition 2.1 (Unfairness). Let C be a classifier trained on dataset $\mathcal{D}$, and f_{min} (respectively f_{maj}) be a statistical measure based on the predictions of C respectively on the minority and majority groups. We define the unfairness of C as $|f_{min} - f_{maj}|$.

This generic definition of unfairness can be instantiated by using any of the statistical measures of fairness mentioned previously. For instance, the *statistical parity* requires that individuals from both the minority and majority groups should have the same probability to be assigned the positive decision. In this context, we define unfairness as:

$$unfairness = \left| \frac{\sum_{x_i \in P_{maj}} pred_i}{|P_{maj}|} - \frac{\sum_{x_i \in P_{min}} pred_i}{|P_{min}|} \right| \quad (1)$$

in which x_i is a sample from the dataset $\mathcal{D}$ and $pred_i$ refers to the prediction of a particular classifier with respect to the decision attribute for x_i.

While our focus is not on individual fairness, we will also use the notion of inconsistency [63] as a way of measuring the resulting individual fairness provided by our contribution. In a nutshell, its purpose is to quantify the tendency of a model to predict the same outcome to similar individuals. We measure it as the average prediction's difference between each instance and its k closest neighbours:

$$inconsistency = \frac{1}{N} \sum_{i=1}^{N} |pred_i - \frac{1}{k} \sum_{x_j \in kNN(x_i)} pred_j| \quad (2)$$

That is, the closest the value of the inconsistency is to zero, the higher the individual fairness.

2.2 Fairness-enhanced learning

While many approaches have been proposed in the literature to enhance the fairness of machine learning methods, they can be categorized in three main families, namely *preprocessing* techniques [11, 23, 30, 63], *algorithmic modification* techniques [10, 31, 32, 62] and *postprocessing* techniques [28]. In a nutshell, preprocessing techniques aim at changing the characteristics of the input data (*e.g.*, by removing existing correlations with the sensitive attribute) so that any classifier trained on this data achieves fairness with respect to its outcome. In contrast, algorithmic modification techniques integrate the fairness constraints directly into a learning algorithm to ensure that the outputted model is fair. Finally, postprocessing techniques modify the outcome of an already trained model to ensure fairness.

Our work falls within the *algorithmic modification* approach in the sense that we propose a fairness-aware algorithm for learning fair rule lists. Existing related works include the seminal work of Calders and Verwer (2010) [10], which consists in training as many classifiers as subgroups considered in the population, using at test time the classifier associated to the corresponding subgroups. Hereafter, we focus on the related work with respect to training a fair *and* interpretable classifier, which is the objective of our method. Kamiran, Calders and Pechenizkiy (2010) [31] have proposed a learning algorithm incorporating the discrimination and accuracy gains into the splitting criterion of a decision tree classifier. In particular, they have devised three combination strategies, namely the difference, the ratio or the sum between accuracy gain and discrimination gain. They have also added a leaf relabeling postprocessing technique that changes the label of selected leaves to improve the fairness.

More recently, Raff, Sylvester and Mills (2018) [52] have applied the difference combination method from [31] to CART decision trees [9] to create both fair decision tree and fair random forest [8]. Zhang and Ntoutsi (2019) [65] improved this combination strategy

Fairness notion	Statistical measure
Statistical parity	Probability of being assigned the positive class
Predictive parity	Positive predictive value
Predictive equality	False positive rate
Equal opportunity	False negative rate
Equalized odds	False negative rate and False positive rate
Conditional use accuracy equality	Positive predictive value and Negative predictive value

Table 1: Summary of some statistical notions of fairness along with the related statistical measure to be equalized.

by introducing the so-called *fair information gain*, which corresponds to the default accuracy gain if there is no unfairness or to the product of both fairness gain and accuracy gain otherwise. In addition, they relied on the use of an Hoeffding tree [19] to provide better fairness and accuracy in the online setting.

Finally, Aghaei, Azizi and Vayanos (2019) [2] proposed `MIP-DT` which adds regularization terms to a mixed-integer programming model to penalize unfairness for creating a generic class of fair decision trees according to individual fairness. Compared to previous techniques modifying the splitting criterion, this technique proposes an exact approach for learning discrimination-aware interpretable model. While our approach for learning fair rule lists is also an exact approach, it differs from [2] on several points. First, `FairCORELS` relies on statistical notions of fairness while `MIP-DT` relies on individual notions of fairness. Moreover, `MIP-DT` achieves good accuracy/fairness trade-offs at the cost of important computational overhead (*i.e.*, $\sim$ 15900 seconds on Adult Income). In contrast, `FairCORELS` achieves similar performances with a smaller computational footprint (*i.e.*, $\sim$ 180 seconds on Adult Income) thanks to the performance of `CORELS`[1].

2.3 Rule list

Rule lists [4, 55] (also known as *decision lists*) are classifiers formed by an ordered list of *if-then* rules with antecedents in the *if* clauses and predictions in the *then* clauses (*e.g.*, Rule lists 1 and 2).

```
if [capital_gain:>7055.5] then (income:>50K)
else if [marital_status:single] then (income:<=50K)
else if [gender:Female] then (income:<=50K)
else if [education:masters_doctorate] then (income:>50K)
else if [education:bachelors] then (income:>50K)
else (income:<=50K)
```

Rule list 1: Example of a rule list that predicts salary category for Adult Income. This rule list found by `FairCORELS` has $unfairness = 0.19$ and $accuracy = 0.8239$.

```
if [capital_gain:>7055.5] then (income:>50K)
else if [marital_status:single] then (income:<=50K)
else if [education:masters_doctorate] then (income:>50K)
else (income:<=50K)
```

Rule list 2: Example of a rule list that predicts salary category for Adult Income. This rule list found by `FairCORELS` has $unfairness = 0.05$ and $accuracy = 0.8078$.

[1]Their experiments were run on a computer with 20 CPUs and 64 GB of RAM. Our experiments were conducted on an Intel Xeon Processor E3-1271 v3 (3.60 GHz) with 32GB of RAM.

More precisely, a rule list $r = (d_p, \delta_p, q_0, K)$ of length $K \geq 0$ is a $(K+1)-$tuple consisting of K distinct association rules $p_k \rightarrow q_k$, in which $p_k \in d_p$ is the antecedent of the association rule and $q_k \in \delta_p$ its associated consequent, followed by a default prediction q_0. Classifying a new data point with a rule list is straightforward. To realize this, the rules are applied sequentially until one rule triggers, in which case the associated prediction is reported. If no rule is triggered, then the default prediction – which typically predicts the majority outcome – is reported. As shown in [55], rule lists generalize decision trees. More precisely for a given size (*i.e.,* the depth of a decision tree or the maximum width of a rule for a rule list), rules lists are strictly more expressive than decision trees. A consequence is that we can obtain more compact models using rule lists, which leads to more interpretability.

`CORELS` [4] is a classification [45] algorithm that enables to learn provably optimal rule lists. `CORELS` represents the search space of rule lists as a trie (*i.e.*, prefixed tree) formed by pre-mined rules from the training data, and uses branch-and-bound techniques to find the optimal rule list. For a given rule list $r = (d_p, \delta_p, q_0, K)$, the objective function to minimize is :

$$R(r, X, Y) = \text{misc}(r, X, Y) + \lambda.K \tag{3}$$

in which $\text{misc}(\cdot) = (1 - \text{accuracy}(\cdot))$ is the classification error, K is the number of rules in prefix r and $\lambda \geq 0$ the regularization parameter used to penalize longer rule lists. The first part of the objective function aims to obtain accurate rule list whereas the second part has for objective to reduce the size of the optimal rule list to limit over-fitting while also increasing interpretability. `CORELS` proposes various search strategies and leverages on a collection of bounds to efficiently prune the search space. Our work is actually an extension of `CORELS` design to ensure fair and accurate models.

2.4 Multi-objective optimization

Optimizing a decision process implies to make a particular choice among a set of alternatives. More precisely, each alternative belongs to the set of feasible solutions and its quality can be assess through a given objective function. In mono-objective optimization, the aim is to find the global optimum, which corresponds to a solution having the best value for this objective function. However, in many applications, one has to consider the concurrent optimization of several objective functions. A multi-objective optimization problem [15, 22] can be defined as follows:

Definition 2.2 (Multi-objective optimization problem).

$$\begin{array}{ll} \textit{minimize} & Z(V) = \{z_1(V), ..., z_m(V)\} \\ \text{subject to} & V \in \Omega \end{array}$$

in which V is the set of decision variables, $m \geq 2$ the number of objective functions, z_i the i^{th} objective function and Ω represents the set of feasible solutions.

Contrary to mono-objective optimization, solving a multi-objective optimization problem produces a set of solutions offering a trade-off between the different objective functions. The comparison between solutions is usually done using a dominance relation, generally the Pareto dominance defined as follows:

Definition 2.3 (Pareto dominance). A solution v dominates a solution v' if v is at least as good as v' for all objectives, and v is strictly better than v' for at least one objective.

The set of non-dominated solutions of a multi-objective optimization problem describes a Pareto frontier.

Various approaches exist for solving multi-objective optimization problems. In this paper, we focus on methods that convert a multi-objective optimization problem into a mono-objective optimization one. They simply consist in combining the different objective functions into a single one before optimizing it. The most basic method is the *weighted-sum-of-objective-functions* in which the objective function to be optimized is a linear combination of the different ones. Another scalar method is the *distance-to-a-reference* method, which consists in defining an "ideal" reference point, encoded as a vector in which each coordinate is the "ideal" value of an objective function. The resulting objective function is the distance between the current objective functions vector and the ideal one. Finally, the last common technique we have used is the ϵ-constraint method, which aims at optimizing only one of the objective functions given a set of constraints on the others. By varying these constraints, we are able to generate the Pareto Front.

3 BI-OBJECTIVE OPTIMIZATION TO LEARN FAIR AND ACCURATE RULE LISTS

We consider the problem of learning fair rule lists. Given a dataset $\mathcal{D} = (X, P, Y)$, our objective is to learn a rule list model characterizing the trade-off between the accuracy and the fairness, or equivalently between the misclassification error and the unfairness. Recall that the objective of the original CORELS method is to minimize the misclassification error misc$(\cdot)$ and the model size K, weighted using a parameter λ.

3.1 An overview of the proposed approach

For taking into account both accuracy and fairness, a straightforward approach is to consider the weighted-sum-of-objective-functions mentioned previously. In this mono-objective approach, the objective of the original CORELS method (Section 2.3) is combined to an unfairness objective (Section 2.1) as follows:

$$\beta \times \text{misc}(r, X, Y) + (1 - \beta) \times \text{unfairness}(r, X) + \lambda.K \qquad (4)$$

in which β is the parameter to control the balance between misclassification error and unfairness. With such approach, one is able to approximate the Pareto front directly by varying β and λ. However, several issues make such an approach unsuitable for our problem. First, combining the two objectives (unfairness with misclassification error) is likely to give coarse results as it approximates both objectives. Second and more importantly, CORELS' original bounds do not hold anymore if the objective function is modified. Indeed, by keeping the same bounds, the optimality of the solution returned is not guaranteed anymore as CORELS' would prune subtrees based on a wrong basis. Finally, when dealing with high fairness values, we observed empirically an uncontrollable behaviour between the regularized sum and the regularization term ($\lambda.K$), leading to very short rule lists.

In addition, we have also evaluated the distance-to-a-reference method in which the "ideal" reference point for misclassification and unfairness is set to $(0, 0)$. However, this approach has the same limitation as the weighted-sum approach due to the invalidity of the CORELS' original bounds in this case.

Our proposition: Pareto Front generation for a given number of points. To create the Pareto Front, we propose an ϵ-constraint method that optimizes the original accuracy objective while respecting a minimum level of fairness. A pseudo code of our method is given in Algorithm 1. This approach allows us to exploit CORELS' original bounds, and to build the frontier of the *accuracy/fairness* trade-offs by varying the constraint on fairness. We use ϵ to control the lower bound of fairness desired in an increasing way. That is, the value 0.0 corresponds to a complete relaxation of fairness (*i.e.*, no minimum fairness is enforced) whereas the value 1.0 for ϵ corresponds to a of 100% of fairness enforced on the training set. Algorithm 1 uses our modified version of CORELS, denoted by FairCORELS and presented in the next section. FairCORELS takes as input the training set $\mathcal{D}_T$ and the fairness parameter ϵ and return an optimal rule list that optimizes the original accuracy objective function within at least $100 \times \epsilon\%$ of fairness. We use a routine $Evaluate(r, \mathcal{D}_{Ev})$ that returns a pair $(fairness, accuracy)$ to evaluate the fairness and accuracy (respectively) of a rule list r of a testing set $\mathcal{D}_{Ev}$.

Algorithm 1 first computes the extreme points, optimizing either fairness or accuracy (Lines 2 to 8), which give the maximum and minimum possible values for fairness. Then, we use these values, and the number of points E, to change ϵ for the minimum amount of fairness enforced at each call in the while loop (Lines 10 to 17). Last, we remove dominated solutions at Line 18.

3.2 FairCORELS

To compute a model optimizing accuracy within a given minimal fairness, a simple adaptation of CORELS would consist in accepting solutions within the branch-and-bound method only when they satisfy the minimal fairness constraint[2]. However, this approach has the main drawback that it leads to useless exploration of the search space as the algorithm is completely uninformed regarding the amount of fairness in the sub-trees it can explore. To avoid these useless computations, we design an unfairness lower bound for CORELS. Our unfairness lower bound relies on the fact that instances captured by the prefix will always be classified similarly for any rule list based on this prefix. Thus, after a prefix evaluation, we compute its unfairness lower bound, which is *the minimum unfairness value that any rule list based on this prefix can have.* If this value is greater than $(1 - \epsilon)$, we prune the entire subtree. In

[2]Our implementation actually *minimizes the misclassification error* with a *maximum acceptable unfairness* (which is equivalent)

Algorithm 1 ParetoFront($\mathcal{D}_T, \mathcal{D}_{Ev}, E, \epsilon$)

Input: $\mathcal{D}_T$: training dataset; $\mathcal{D}_{Ev}$: testing dataset; E : Number of points of the Pareto Front to be calculated; ϵ: fairness parameter;
Output: L : List of non-dominated $(fairness, accuracy)$ tuples

$L \leftarrow []$
$r \leftarrow$ FairCORELS$(\mathcal{D}_T, \epsilon = 1.0)$
$(fairness_0, accuracy_0) \leftarrow Evaluate(r, \mathcal{D}_{Ev})$
$(fairness_{T0}, accuracy_{T0}) \leftarrow Evaluate(r, \mathcal{D}_T)$
$L \leftarrow L \cup \{(fairness_0, accuracy_0)\}$
$r \leftarrow$ FairCORELS$(\mathcal{D}_T, \epsilon = 0.0)$
$(fairness_1, accuracy_1) \leftarrow Evaluate(r, \mathcal{D}_{Ev})$
$(fairness_{T1}, accuracy_{T1}) \leftarrow Evaluate(r, \mathcal{D}_T)$
$L \leftarrow L \cup \{(fairness_1, accuracy_1)\}$
$\Delta \leftarrow \frac{fairness_{T0} - fairness_{T1}}{E}$
$f \leftarrow fairness_{T0}$
while $f > fairness_{T1}$ **do**
  $r \leftarrow$ FairCORELS$(\mathcal{D}_T, \epsilon = f)$
  $(f, accuracy) \leftarrow\leftarrow Evaluate(r, \mathcal{D}_{Ev})$
  $L \leftarrow L \cup \{(f, accuracy)\}$
  $f \leftarrow f - \Delta$
end while
$L \leftarrow eliminateDominatedSolutions(L)$
return L

practice, this new bound leads to rule lists with a better quality in shorter computational time as demonstrated later in Section 4.2.

The details of the bound computation are described in Algorithm 2. In this algorithm, we compute in Line 1 a vector of Booleans *capt* in which $capt_i$ is true if and only if the data point x_i is captured by the prefix (*i.e.*, x_i is classified by a rule contained in the prefix and does not fall into the default decision). In Line 2, we also compute a vector of Booleans *pred* in which $pred_i$ is true if and only if the data point x_i is classified positively by the prefix. The rest of the algorithm is straightforward as it computes exactly how much unfairness is contained in the prefix.

FairCORELS, our new version of CORELS, is depicted in Algorithm 3. We refer the reader to the original CORELS papers for a detailed understanding of the algorithm [3, 4]. The integration of the fairness lower bound is performed at Line 13 before exploring any subtree. We use $u(r, X, Y)$ (respectively $b(r, X, Y)$) to denote an oracle that measures the unfairness (respectively accuracy objective function) of a rule list r given X and Y.

4 EXPERIMENTS

In this section, we report on the experiments that we have conducted to evaluate (1) the impact of the unfairness lower bound, (2) the accuracy/fairness trade-offs provided by FairCORELS using Algorithm 1, for different datasets using the statistical parity as the fairness notion and (3) the performance of FairCORELS compared to existing solutions, namely FairForest [52] and FAHT [65].

4.1 Experimental setting

Description of datasets. We conduct our experiments on four public datasets that are extensively used in the fairness literature due

Algorithm 2 $ulb(r, \mathcal{D}_T)$: Unfairness lower bound computation

Input: r : prefix of rules; $\mathcal{D}_T = (X, P, Y)$: training dataset ;
Output: lb : Unfairness lower bound

1: $capt \leftarrow Compute\_capt(r, \mathcal{D}_T)$ ▷ captured instances vector
2: $pred \leftarrow Compute\_pred(r, \mathcal{D}_T)$ ▷ prediction vector
3: $MinScore_{maj} \leftarrow \frac{\sum_{x_i \in P_{maj}} pred_i \times capt_i}{|P_{maj}|}$
4: $MaxScore_{maj} \leftarrow \frac{|P_{maj}| - \sum_{x_i \in P_{maj}} (\neg pred_i) \times capt_i}{|P_{maj}|}$
5: $MinScore_{min} \leftarrow \frac{\sum_{x_i \in P_{min}} pred_i \times capt_i}{|P_{min}|}$
6: $MaxScore_{min} \leftarrow \frac{|P_{min}| - \sum_{x_i \in P_{min}} (\neg pred_i) \times capt_i}{|P_{min}|}$
7: **if** $MaxScore_{min} < MinScore_{maj}$ **then**
8: $lb \leftarrow MinScore_{maj} - MaxScore_{min}$
9: **else if** $MaxScore_{maj} < MinScore_{min}$ **then**
10: $lb \leftarrow MinScore_{min} - MaxScore_{maj}$
11: **else**
12: $lb \leftarrow 0$
13: **end if**
14: **return** lb

Algorithm 3 FairCORELS$(\mathcal{D}_T, \epsilon)$

Input: $\mathcal{D}_T = (X, P, Y)$: training dataset; ϵ: fairness parameter;
Output: r^c: Rule list r^c with minimum objective z^c such that $u(r^c, X, Y) \leq \epsilon$

1: $(r^c, z^c, S) \leftarrow Pre\text{-}processing()$ ▷ Pre-processing to initialize best rule list r^c and objective z^c and to compute a set of pre-mined rules S
2: $Q \leftarrow queue([()])$ ▷ Initialize queue with empty prefix
3: **while** Q not empty **do**
4: $r \leftarrow Q.pop()$ ▷ Remove prefix r from the queue
5: $u_r \leftarrow u(r, X, Y)$ ▷ Compute unfairness of r
6: **if** $b(r, X, Y) < z^c$ **then** ▷ **Objective lower bound**
7: $z \leftarrow R(r, X, Y)$
8: **if** $z < z^c$ and $u_r \leq (1 - \epsilon)$ **then**
9: $(r^c, z^c) \leftarrow (r, z)$ ▷ Update best rule list and objective
10: **end if**
11: **for** s in S **do** ▷ **Branch**: Enqueue r's children
12: **if** s not in r **then**
13: **if** $ulb(r \cup \{s\}, \mathcal{D}_T) \leq (1 - \epsilon)$ **then** ▷ Apply the unfairness pruning
14: $Q.push((r, s))$
15: **end if**
16: **end if**
17: **end for**
18: **end if**
19: **end while**
20: **return** r^c

to their biased nature, namely *COMPAS* [5], *Adult Income* [24], *German Credit* [24] and *Default Credit* [24]. In a nutshell, the COMPAS dataset gathers 6,167 records from criminal offenders in Florida during 2013 and 2014 each described by 11 attributes with the classification task considered being to predict whether or not a subject will reoffend within two years after the screening. We use the race (*i.e.*, african-american or caucasian) as the sensitive attribute. The Adult Income dataset contains information about more than 45,000 individuals from the 1994 U.S. census, each described by 14 attributes. For this dataset, the sensitive attribute is the gender while the objective of the decision task is to predict whether an individual makes more or less than 50,000$ per year in terms of income. The German Credit dataset aims to classify people according to whether or not they have a good or bad credit risk. This dataset contains 1,000 individuals, each described by 20 attributes with the age range of the individual (over or below 25 years old) being the sensitive attribute. Finally, the Default Credit dataset is composed of information of 30,000 Taiwanese credit card users and aims to classify whether a user will default with the gender being the sensitive attribute. A detailed summary of these datasets is presented in Table 2. For each dataset, we report the size of the dataset (*i.e.*, the number of data points), the dimensionality in terms of number of attributes and the number of binary attributes, the sensitive attribute and decision attribute.

Setup. We use both the unfairness and accuracy to evaluate `FairCORELS`. In particular, we rely on statistical parity as described in Section 2.1 as the unfairness metric. To compare with results found in `FairForest` [52], we also use the *inconsistency* metric (also defined in Section 2.1) as well as the *delta*, which is defined as follows.

$$delta = accuracy - unfairness \tag{5}$$

To build the Pareto front for each dataset, we use Algorithm 1. In particular, for each step of the algorithm, we run `FairCORELS` with a 10-fold cross validation and report the average values for the accuracy, the unfairness, the delta as well as the inconsistency. To select the value of λ yielding to better performance, we run an hyperparameter search on each fold, using `HyperOpt`[3]. This enables us to observe that in practice λ does not vary too much when using different values for ϵ. For instance on Adult Income, the average (respectively standard deviation) of λ is $3e-3$ (respectively $1e-3$). Our experiments were conducted on an Intel Xeon Processor E3-1271 v3 (3.60 GHz) with 32GB of RAM. `FairCORELS` is implemented in C++ and based on the original source code of CORELS[4].

4.2 Results

Impact of the unfairness lower bound. Before exposing our main results with respect to the trade-off between fairness and accuracy, we have assessed the impact of the unfairness lower bound presented in Section 3 experimentally. To realize this, we first evaluated `FairCORELS` with several branching strategies using best-first search and breadth-first search (BFS). We have observed that BFS outperforms the best-first search when dealing with high number of attributes. While it is clear that rule lists with smaller depths should be explored first while using BFS, the order in which the rule lists with the same depths should be explored is not obvious. Therefore, we have tested several customized BFS strategies when breaking ties between the best two prefixes (left and right nodes) of the priority queue with a same length. These different strategies are depicted below:

- `CORELS`*'s original BFS*: choose the right node.
- *FIFO BFS*: choose the node that was inserted first in the queue.
- *Objective-aware BFS*: choose the node with the best objective function.
- *Lower-bound aware BFS*: choose the node with the best lower bound.
- *Random BFS*: break ties at random.

In Table 3, we report the results using the Adult Income dataset for an unfairness parameter of 0.01 by setting the maximum number of nodes in the trie to 10^7. For each strategy, we describe the number of explored nodes before finding the best solution (*i.e.*, Nodes), the accuracy of that solution on the test set (*i.e.*, Accuracy) and the number of subtrees pruned when using the unfairness lower-bound. These results clearly show the benefits of using our unfairness lower bound as it finds better solutions quicker by pruning a considerable portion of the search tree.

Unfairness and accuracy trade-offs. We use Algorithm 1 to compute the set of non-dominated solutions in terms of both unfairness and misclassification error minimization. In particular, we use the objective-aware BFS (*cf.* Table 3) exploration strategy. The Pareto frontiers in Figure 1 show, for each dataset, the accuracy-fairness trade-offs achieved by `FairCORELS`. For instance, on Adult Income, `FairCORELS` is able to reduce the unfairness from 0.19 to 0.01 while only increasing the error from 0.17 to 0.20. Figure 2 gives a close-up look at the tradeoffs achieved by `FairCORELS` on Adult Income for $\epsilon >= 0.95$. As suggested in recent works [34, 35], providing the Pareto frontier instead of a unique model gives the possibility to the stakeholders or policymakers to select the right model in a domain-specific manner.

Comparison with others methods. To compare our method with other fairness-enhanced approaches [52, 65], we have selected 5 points along the Pareto fronts of both German Credit and Adult Income datasets: the rule list with the best (1) accuracy, (2) unfairness, (3) delta, (4) accuracy for $\epsilon >= 0.95$ and (5) accuracy for $\epsilon >= 0.99$. We also apply the same analysis to the COMPAS and Default Credit datasets.

As shown in Table 4, `FairCORELS` achieves better unfairness/accuracy trade-offs compared to both `FairForest` and `FAHT`. In particular, `FairCORELS` achieves an unfairness of 0.0493 for an accuracy of 0.8115 on Adult Income. In comparison, `FAHT` achieves an unfairness of 0.1629 for an accuracy of 0.8183 on the same dataset. One of the strength of `FairCORELS` is that it allows to efficiently compute the Pareto front, thus allowing to easily visualize the achievable trade-offs. In addition, since the obtained models are interpretable, we can inspect them to decide whether or not they are legitimate or relevant. For instance, on German Credit, when the fairness requirement is high ($\epsilon >= 0.99$), the accuracy of the solution found by

[3] https://github.com/hyperopt/hyperopt
[4] https://github.com/nlarusstone/corels

Dataset	#Size	#Attributes (#Bin. attr.)	Sensitive Attribute	Decision attribute
COMPAS	6,167	11/25	Race: African-American/Caucasian	2 years recidivism
Adult Income	45,222	14/69	Gender: Male/Female	Income $\geq$ 50k
German Credit	1,000	20/80	Age : $\geq$ 25	Good/Bad credit score
Default Credit	30,000	25/90	Gender : Male/Female	Default of payment

Table 2: Summary of the datasets used.

	Without unfairness bound		With unfairness bound		Subtrees pruned with the bound
	Nodes	Accuracy	Nodes	Accuracy	
Original CORELS	18, 365, 708	79.17%	16, 523, 605	79.25%	233, 631
FIFO	1, 446, 553	78.68%	1, 439, 512	78.68%	282, 272
Objective-aware	6, 478, 865	(79.25%)	6, 232, 246	79.25%	532, 822
Lower-bound aware	3, 119, 577	(78.68%)	3, 103, 109	78.68%	363, 866
Random	5, 447, 942	(79.17%)	4, 453, 465	79.25%	212, 324

Table 3: Impact of the unfairness lower-bound on different BFS strategies.

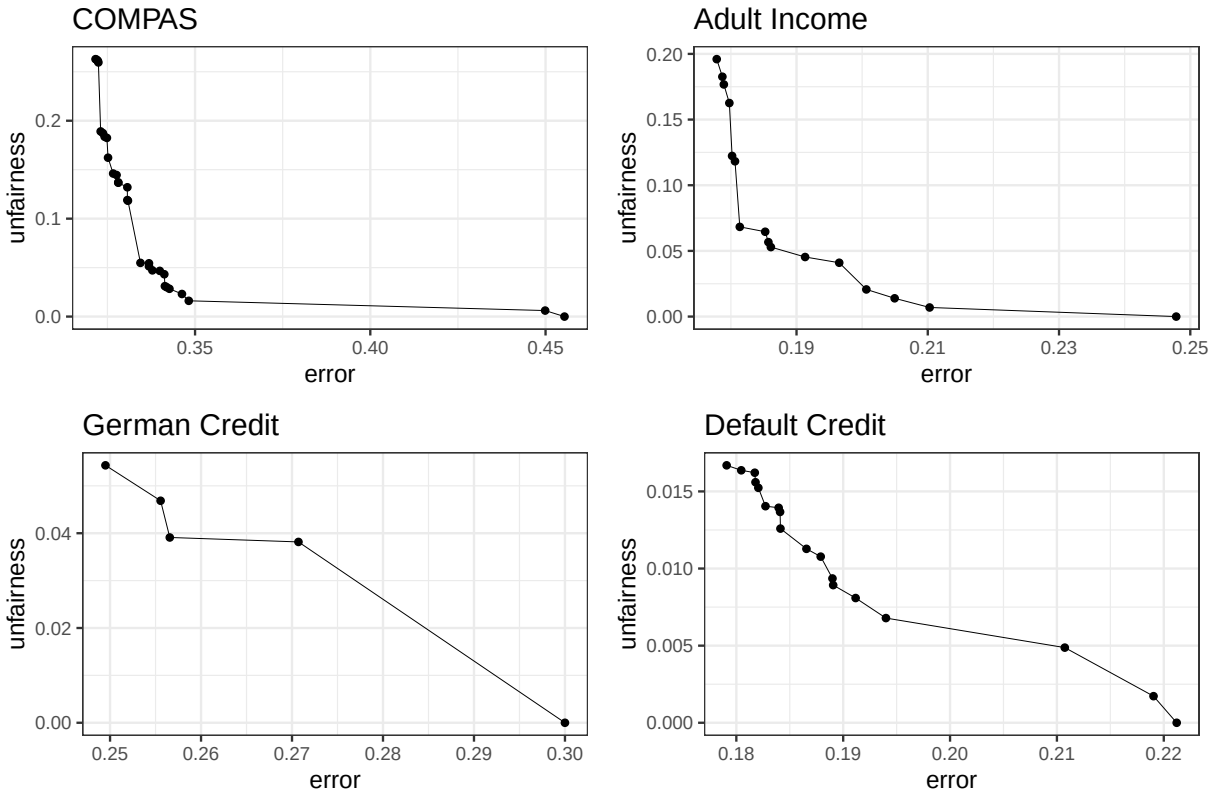


Figure 1: Pareto front (unfairness, misclassification error) for COMPAS, Adult Income, German Credit and Default Credit.

`FairCORELS` is not better than predicting the majority class. However, by slightly relaxing the fairness constraint, we able to find meaningful models with better accuracy.

Table 5 shows that `FairCORELS` also achieves good performances on both COMPAS and Default Credit. In addition, Figure 3 shows for the model with best *accuracy* (respectively best *Delta*) on COMPAS dataset, the relative importance of the attributes on the prediction outcome as well as the description of the corresponding models. We used the `FairML` [1] library to audit the models we produced. For a given rule list, `FairML` shows for each feature, its relative importance in the decision-making process. We can clearly see that, in the models with higher accuracy, the sensitive attributes are

		German Credit				Adult Income			
		Accuracy	Delta	Unfairness	Inconsistency	Accuracy	Delta	Unfairness	Inconsistency
FairCORELS	Best accuracy	**0.7505**	0.6962	0.0543	0.2045	**0.8225**	0.6238	0.1987	0.0496
	Best unfairness	0.7000	0.7000	**0.0000**	**0.0000**	0.7521	0.7521	**0.0000**	**0.0000**
	Best delta	0.7434	**0.7043**	0.0391	0.1936	0.7928	**0.7845**	0.0083	0.0285
	Best accuracy for $\epsilon >= 0.95$	0.7444	0.6976	0.0469	0.1966	0.8115	0.7621	0.0494	0.0307
	Best accuracy for $\epsilon >= 0.99$	0.7000	0.7000	**0.0000**	**0.0000**	0.7928	**0.7845**	0.0083	0.0285
Fair Forest	Decision tree	0.6990	0.6908	0.0082	0.0070	0.7511	0.7444	0.0067	0.0033
	Random Forest	0.7000	0.7000	**0.0000**	**0.0000**	0.7530	0.7530	**0.0000**	**0.0000**
FAHT		-	-	-	-	0.8183	0.6554	0.1629	-

Table 4: Results of `FairCORELS` on German Credit and Adult Income and comparison with existing solutions.

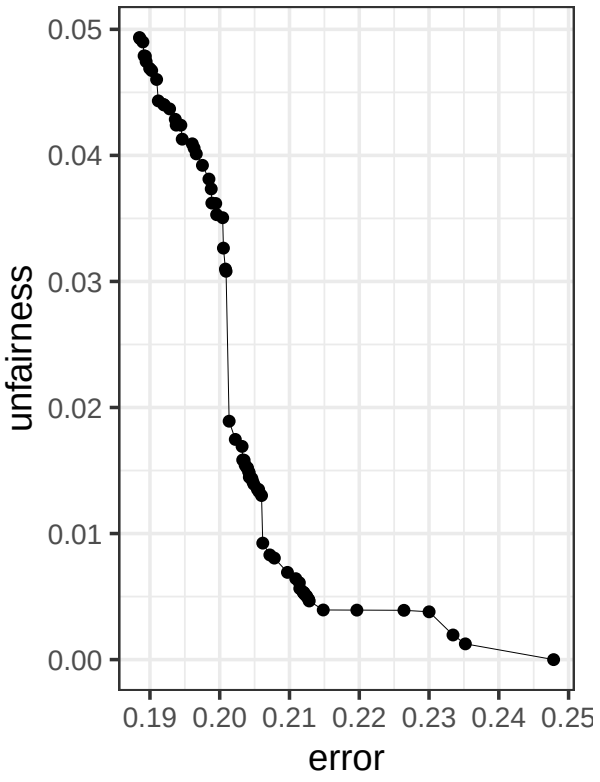


Figure 2: Accuracy-fairness tradeoffs of `FairCORELS` on Adult Income, for $\epsilon >= 0.95$

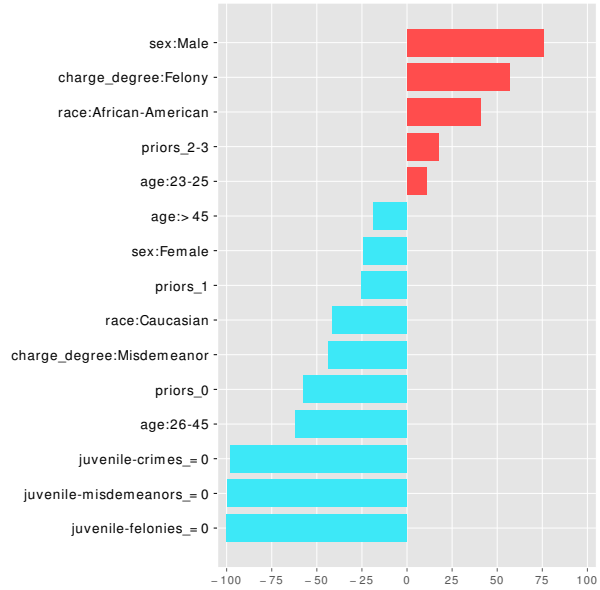
used in a discriminatory manner. In the models with higher *Delta*, a form of positive discrimination, favouring individuals that were originally discriminated, can be observed.

5 CONCLUSION

In this paper, we presented `FairCORELS`, a fairness-aware algorithm to learn fair interpretable models by design. We formulated the problem of learning fair rule lists as a bi-objective formulation of the problem of learning rule list, in which we jointly minimize the unfairness as well as the classification error. The proposed `FairCORELS` algorithm is embedded in a bi-objective optimization method to compute the set of non-dominated solutions. We also design a new lower bound for the unfairness objective to prune the search space efficiently, and a collection of branching strategies to improve the effectiveness of the search for fair rule lists. Our experiments show that this technique aims at finding better fairness/accuracy trade-offs. While in this work, we have focused on statistical parity, as part of our future work, we will design additional unfairness lower bounds for other statistical notions of fairness. In addition, we envision to integrate individual notions of fairness in our framework.

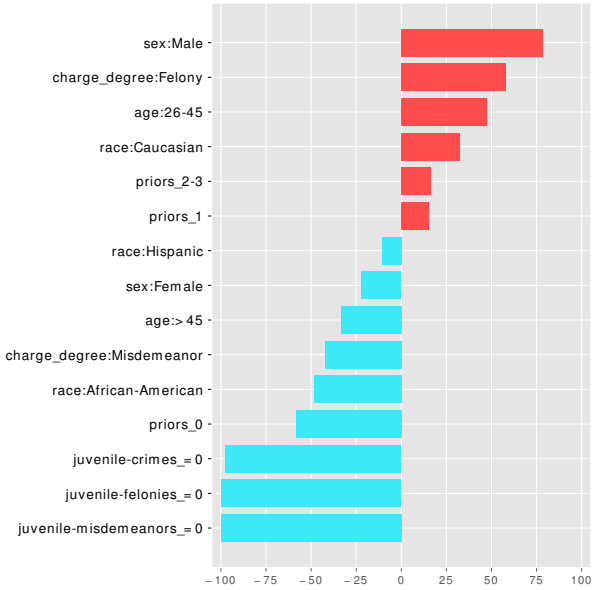
		COMPAS				Default Credit			
		Accuracy	Delta	Unfairness	Inconsistency	Accuracy	Delta	Unfairness	Inconsistency
FairCORELS	Best accuracy	**0.6777**	0.4165	0.2612	0.1170	**0.8209**	**0.8042**	0.0167	0.0597
	Best unfairness	0.5446	0.5446	**0.0000**	**0.0000**	0.7788	0.7788	**0.0000**	**0.0000**
	Best delta	0.6518	**0.6357**	0.0161	0.1189	**0.8209**	**0.8042**	0.0167	0.0597
	Best accuracy for $\epsilon >= 0.95$	0.6648	0.6201	0.0447	0.1359	**0.8209**	**0.8042**	0.0167	0.0597
	Best accuracy for $\epsilon >= 0.99$	0.5552	0.5520	0.0032	0.0139	0.8106	0.8010	0.0095	0.0577

Table 5: Results of `FairCORELS` on COMPAS and Default Credit.




```
if [priors:>3] then recidivism
else if [juvenile-felonies:>0] then recidivism
else if [age:26-45] then not recidivism
else if [age:>45] then not recidivism
else if [priors:2-3] then recidivism
else if [age:18-20] then recidivism
else if [race:Caucasian] then not recidivism
else if [juvenile-crimes:>0] then recidivism
else if [priors:0] then not recidivism
else if [sex:Female] then not recidivism
else recidivism
```

Rule list 3: Best accuracy on COMPAS.




```
if [age:>45] then not recidivism
else if [priors:0] then not recidivism
else if [race:Caucasian] then recidivism
else if [priors:>3] then recidivism
else if [age:26-45] then not recidivism
else recidivism
```

Rule list 4: Best delta on COMPAS.

Figure 3: Models with best accuracy and best delta on COMPAS.

REFERENCES

[1] Julius A Adebayo et al. 2016. *FairML: ToolBox for diagnosing bias in predictive modeling*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[2] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. 2019. Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making. *arXiv preprint arXiv:1903.10598* (2019).

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

[4] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. 2018. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research* 18, 234 (2018), 1–78.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. *ProPublica, May* 23 (2016).

[6] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* 0, 0 (2018), 0049124118782533.

[7] Nick Bostrom and Eliezer Yudkowsky. 2014. The ethics of artificial intelligence. *The Cambridge handbook of artificial intelligence* 316 (2014), 334.

[8] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[9] Leo Breiman. 2017. *Classification and regression trees*. Routledge.

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

[11] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*. 3992–4001.

[12] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. 2019. Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*. 319–328.

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

[14] Alexandra Chouldechova and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).

[15] Yann Collette and Patrick Siarry. 2013. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media.

[16] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic Decision Making and the Cost of Fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, 797–806. https://doi.org/10.1145/3097983.3098095

[17] Mark Craven and Jude W Shavlik. 1996. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*. 24–30.

[18] Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. 2018. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*. 4655–4665.

[19] Pedro M. Domingos and Geoff Hulten. 2000. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, Raghu Ramakrishnan, Salvatore J. Stolfo, Roberto J. Bayardo, and Ismail Parsa (Eds.). ACM, 71–80. https://doi.org/10.1145/347090.347107

[20] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

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

[22] Matthias Ehrgott. 2005. *Multicriteria optimization*. Vol. 491. Springer Science & Business Media.

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

[24] Andrew Frank and Arthur Asuncion. 2010. UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California. *School of information and computer science* 213 (2010), 2–2.

[25] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 329–338.

[26] B Goodman and SR Flaxman. 2017. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38, 3 (2017), 50–57.

[27] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)* 51, 5 (2018), 93.

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

[29] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. 2016. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*. 325–333.

[30] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[31] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*. IEEE, 869–874.

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

[33] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 643–650.

[34] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In *International Conference on Machine Learning*. 2569–2577.

[35] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2019. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 100–109.

[36] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*. 656–666.

[37] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2017. Human decisions and machine predictions. *The quarterly journal of economics* 133, 1 (2017), 237–293.

[38] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[39] Hian Chye Koh, Wei Chin Tan, and Chwee Peng Goh. 2006. A two-step method to construct credit scoring models with data mining techniques. *International Journal of Business and Information* 1, 1 (2006).

[40] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*. 4066–4076.

[41] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2017. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154* (2017).

[42] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. 2017. Fair, Transparent, and Accountable Algorithmic Decision-making Processes. *Philosophy & Technology* (2017), 1–17.

[43] Jiuyong Li, Hong Shen, and Rodney Topor. 2002. Mining the optimal class association rule set. *Knowledge-Based Systems* 15, 7 (2002), 399–405.

[44] Zachary C. Lipton. 2018. The mythos of model interpretability. *Commun. ACM* 61, 10 (2018), 36–43. https://doi.org/10.1145/3233231

[45] Bing Liu, Wynne Hsu, Yiming Ma, et al. 1998. Integrating classification and association rule mining.. In *KDD*, Vol. 98. 80–86.

[46] Tyler McCormick, Cynthia Rudin, and David Madigan. 2011. A hierarchical model for association rule mining of sequential events: An approach to automated medical symptom prediction. (2011).

[47] Claire Cain Miller. 2015. Can an Algorithm Hire Better Than a Human? https://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html

[48] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* 73 (2018), 1–15.

[49] Razieh Nabi and Ilya Shpitser. 2018. Fair inference on outcomes. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[50] Arvind Narayanan. 2018. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*.

[51] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, Joao Marques-Silva, and IS RAS. 2018. Learning Optimal Decision Trees with SAT.. In *IJCAI*. 1362–1368.

[52] Edward Raff, Jared Sylvester, and Steven Mills. 2018. Fair forests: Regularized tree induction to minimize model bias. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 243–250.

[53] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.

[54] Peter R Rijnbeek and Jan A Kors. 2010. Finding a short and accurate decision rule in disjunctive normal form by exhaustive search. *Machine learning* 80, 1 (2010), 33–62.

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

[56] Naeem Siddiqi. 2012. *Credit risk scorecards: developing and implementing intelligent credit scoring*. Vol. 3. John Wiley & Sons.

[57] Berk Ustun and Cynthia Rudin. 2016. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* 102, 3 (2016), 349–391.

[58] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 1–7.

[59] Fulton Wang and Cynthia Rudin. 2015. Falling rule lists. In *Artificial Intelligence and Statistics*. 1013–1022.

[60] Rebecca Wexler. 2017. When a computer program keeps you in jail: How computers are harming criminal justice. *New York Times* (2017).

[61] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. 2017. Scalable Bayesian rule lists. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3921–3930.

[62] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1171–1180.

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

[64] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. 2017. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180, 3 (2017), 689–722.

[65] Wenbin Zhang and Eirini Ntoutsi. 2019. FAHT: An Adaptive Fairness-aware Decision Tree Classifier. *arXiv preprint arXiv:1907.07237* (2019).

[66] Indre Zliobaite. 2015. On the relation between accuracy and fairness in binary classification. *arXiv preprint arXiv:1505.05723* (2015).