Fairness-Aware Decision Tree Editing Based on Mixed-Integer Linear Optimization

Kentaro Kanamori
Graduate School of Information Science and Technology, Hokkaido University
kanamori@ist.hokudai.ac.jp

Hiroki Arimura
(affiliation as previous author)
arim@ist.hokudai.ac.jp

keywords: fairness in machine learning, decision trees, mixed-integer linear optimization

Summary

In the application of machine learning models to decision-making tasks (e.g., loan approval), fairness of their predictions has emerged as an important topic in recent years. If decision-makers detect unfairness in their models during deployment, they must modify the models to satisfy constraints on a specific discrimination criterion. However, simply retraining a model from scratch under fairness constraints may raise serious reliability issues caused by differences in prediction and interpretation between the initial model and retrained model. In this paper, we propose a post-processing framework, named Fairness-Aware Decision tree Editing (FADE), that converts a given biased decision tree into a fair decision tree without significantly changing it in terms of its prediction and interpretation. For this purpose, we introduce two dissimilarity measures between decision trees based on the prediction discrepancy and edit distance. We propose a mixed-integer linear optimization formulation for minimizing the dissimilarity measures under fairness constraints. Numerical experiments on real datasets demonstrate the effectiveness of our method in comparison with existing methods.

1. Introduction

In the application of machine learning models to decision-making tasks, such as loan approval, there have been increasing concerns about the fairness of the predictions [Hajian 16]. If the predictions are unfair, which means that there is disparity of the predictions between groups defined by sensitive attributes (e.g., gender or race) [Barocas 16], the models are no longer usable in decision-making tasks, even if they achieve high accuracy. To avoid such disparity of predictions and achieve model fairness, a number of methods for evaluating the disparity [Adler 18, Calders 09, Hardt 16, Zafar 17] and training fair models [Kamiran 10, Kamishima 12, Zafar 19] have been proposed.

If decision-makers identify unfairness in their deployed model, they must modify the model so as to satisfy constraints on specific discrimination criteria, such as demographic parity (DP) [Calders 09]. However, in various situations of real applications (e.g., dataset shift [Quionero-Candela 09]), there still remain many difficulties in eliminating the disparity from deployed models [Wang 19]. In the following, we demonstrate that simply retraining a model from scratch under fairness constraints raises serious reliability issues. We consider an out-of-sample deployment scenario [Ustun 19b], which is a special case of the dataset shift, on the German dataset [Dua 17]. A decision-maker deploys the decision tree presented in Figure 1(a) to predict whether individuals will default their loan. However, because the training dataset does not include individuals under 25 years old, the decision tree turns out to be unfair to young individuals in terms of DP. The decision-maker then retrains it from scratch to be fair with respect to individuals’ age using an existing method [Kamiran 10]. Although the retrained decision tree presented in Figure 1(b) achieves lower DP than the initial decision tree, we argue that there are serious reliability issues from the following two perspectives:

• **Prediction**: To satisfy fairness constraints, the prediction results of some individuals must be changed. Previous research indicated a risk of prediction conflicts between an initial model and retrained model [Marx 20, Ustun 19a]. In Figure 1, 26% of individuals receive different prediction results from the retrained decision tree than the initial decision tree.

• **Interpretation**: There are often representational differences between the initial decision tree and retrained decision tree, which implies differences in their interpretations. These differences may cause problems re-
Regarding the vulnerability of model interpretation [Guidotti 19], such as fairwashing [Aivodji 19]. In Figure 1, the prediction rules of four out of seven nodes in the retrained decision tree differ from those of the initial decision tree. Based on the above observations, we aim to edit a given biased decision tree $h$ into a fair decision tree $h^*$ without significantly changing it in terms of its prediction and interpretation. Roughly speaking, we consider the following optimization problem:

$$\min_{h^*} \Gamma(h, h^*) \text{ subject to } \Delta(h^*) \leq \theta,$$

where $\Gamma$ is a dissimilarity measure between decision trees, $\Delta$ is a discrimination criterion, and $\theta$ is a discrimination threshold. Our research goals are (i) to introduce dissimilarity measures $\Gamma$ between decision trees based on the above two perspectives, and (ii) to develop a method for minimizing $\Gamma(h, h^*)$ under fairness constraints on $\Delta(h^*)$.

### 1.1 Our Contributions

In this paper, we propose a post-processing framework, named Fairness-Aware Decision tree Editing (FADE), that edits the prediction rules of nodes of a given trained decision tree to satisfy fairness constraints without significantly changing the decision tree. Our contributions can be summarized as follows:

- To evaluate the degree of change from a given decision tree to its edited one, we introduce two dissimilarity measures between decision trees based on the prediction discrepancy [Marx 20] and edit distance [Tai 79, Zhang 89].

- We formulate the problem of finding an optimal edit operation according to the proposed dissimilarity measures under fairness constraints as a mixed-integer linear optimization (MILO) problem, which can be efficiently solved by off-the-shelf MILO solvers, such as CPLEX [IBM 20].

- By experiments on real datasets, we confirmed that decision trees edited by our method achieve comparable accuracy and fairness with a few edits to that of decision trees trained by an existing method for learning fair decision trees [Kamiran 10].

Figure 1(c) presents a decision tree edited by the proposed FADE method. The ratio of conflicting predictions is reduced to 15%, and only one node in the initial decision tree (Figure 1(a)) is edited. This result suggests that (i) FADE can transform a biased decision tree into a fair one with several changes in terms of its prediction and interpretation, and (ii) the edited decision tree achieves comparable accuracy and fairness to the retrained decision tree.

### 1.2 Related Work

**Fairness in Machine Learning.** According to [Hajian 16], existing methods for achieving model fairness can be divided into three groups: pre-processing [Calmon 17, Kamiran 12], in-processing [Kamiran 10, Kamishima 12, Zafar 19], and post-processing [Calders 10, Wang 19] approaches. Our method is most closely related to post-processing approaches, which adjust already trained models under fairness constraints based on a discrimination criterion, such as demographic parity [Calders 09] or equal opportunity [Hardt 16]. In particular, [Kamiran 10] proposed a post-processing algorithm that modifies the predictive labels of leaves of a trained decision tree to satisfy fairness constraints. Our approach can be regarded as an extension of their approach, as we edit not only leaves, but also internal nodes.

**Decision Trees.** With respect to interpretability, a decision tree is one of the most interpretable model types. Top-down greedy approaches (e.g., CART [Breiman 84]) are popular methods for training decision trees. In contrast, several non-greedy methods that are either based on MILO [Aghaei 19, Bertsimas 17, Verwer 19] or branch-and-bound algorithms [Aglin 20, Hu 19] have recently been proposed. Our approach is based on MILO methods, and can be regarded as their extension to determine how to edit an already trained decision tree to be fair. It should also be noted that [Carreira-Perpiñán 18] proposed an editing methods for decision trees that aims to enhance their accuracy and interpretability. However, in contrast, our method aims to enhance their fairness.

### 2. Preliminaries

For a positive integer $n \in \mathbb{N}$, we write $[n] := \{1, \ldots, n\}$. For a proposition $\psi$, $I[\psi]$ denotes the indicator of $\psi$; that is, $I[\psi] = 1$ if $\psi$ is true, and $I[\psi] = 0$ if $\psi$ is false.
where holds since

where

order traversal. Let

Let a tuple

sensitive attribute

sensitive attribute

x

put vector

gender or race [Hajian 16]. Let a tuple

of each internal node. Figure 2 presents an

equal opportunity [Hardt 16]. The values of these crite-

rion based on statistical parity. Our method can also be

extended to other criteria based on statistical parity, such as equal opportunity [Hardt 16]. The values of these crite-

rion approach 1 if the predictions of h tend to be unfair for

z, and they approach 0 if the predictions tend to be fair.

We say that a classifier h satisfies DP if

where is the probability over the joint distribution on

(\(z, h(x)\)). Because we cannot observe, we instead define the DP score \(\Delta: \mathcal{H} \rightarrow [0, 1]\) using the empirical probability \(\hat{P}\) on a given sample

\[\Delta(h \mid S) := \left| \hat{P}(h(x) = 1 \mid z = 1) - \hat{P}(h(x) = 1 \mid z = 0) \right|\].

3. Problem Statement

In this chapter, we formally define our Fairness-Aware Decision tree Editing (FADE) problem.

3.1 Dissimilarity Measure Between Decision Trees

To evaluate the dissimilarity between a given decision tree and its edited one in terms of its prediction and interpre-

tation, we introduce two dissimilarity measures \(\Gamma: \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}_{\geq 0}\) for decision trees.

\(\S\).

\S 1 Prediction Discrepancy

One dissimilarity measure is the prediction discrepancy (PD) proposed by [Marx 20] to evaluate predictive multiplicity in a classification task. PD represents the ratio of individuals whose prediction results are changed if the initial classifier \(h_1\) is replaced by another classifier \(h_2\). In equa-

tion, PD between \(h_1\) and \(h_2\) is defined as follows:

\[
\Gamma_{\text{PD}}(h_1, h_2 \mid S) = \frac{1}{|S|} \sum_{(x,y,z) \in S} \mathbb{1}[h_1(x) \neq h_2(x)],
\]

where S is a given sample. PD can be regarded to evaluate the functional dissimilarity between two classifiers on S.

\S 2 Edit Distance

The second dissimilarity measure is the edit distance (ED) [Tai 79, Zhang 89], which is a standard dissimilarity measure between labeled ordered trees. The ED between two labeled ordered trees \(T_1, T_2\) is defined as the minimal length of the sequence of edit operations required to transform \(T_1\) into \(T_2\). Three edit operations are available: insertion, deletion, and relabeling.
By regarding the branching features \(d_i \in [D]\) and predictive labels \(\hat{y}_i \in \mathcal{Y}\) as node labels, decision trees can be viewed as labeled ordered binary trees with node labels \(\Sigma = [D] \cup \mathcal{Y}\). Let \(e(h_1, h_2) = (e_1, \ldots, e_M)\) be a sequence of edit operations to transform \(h_1\) into \(h_2\). Then, the ED between \(h_1\) and \(h_2\) is defined as follows:

\[
\Gamma_{ED}(h_1, h_2) = \min_{e(h_1, h_2)} \left( \sum_{e \in e(h_1, h_2)} \text{cost}(e) \right),
\]

where \(\text{cost}(e) \geq 0\) is the cost of the edit operation \(e\).

Let \(\epsilon\) be a special blank symbol not in \(\Sigma\), and let \(\Sigma_\epsilon = \Sigma \cup \{\epsilon\}\). Then, each edit operation \(e\) corresponding to nodes \(t_1\) and \(t_2\) can be expressed as a pair of their labels \((s_1, s_2)\), where \(s_1, s_2 \in \Sigma \cup \{\epsilon\}\) are the node labels of \(t_1\) and \(t_2\), respectively [Kondo 14]. We can express \(e\) by (i) \((\epsilon, s_2)\) for inserting a node with a label \(s_2\), (ii) \((s_1, \epsilon)\) for deleting a node with a label \(s_1\), and (iii) \((s_1, s_2)\) for relabeling a node with a label \(s_1\) by another label \(s_2\). We define the cost of \(e\) by \(\text{cost}(e) = \text{dist}(t_1, t_2)\), where \(\text{dist}(t_1, t_2)\) is some dissimilarity measure between the labels of nodes \(t_1\) and \(t_2\). In this paper, for simplicity, we assume that the unit cost \(\text{dist}(t_1, t_2) = I[s_1 \neq s_2]\).

### 3.2 Edit Operation for Decision Tree

As edit operations for a given initial decision tree \(h \in \mathcal{H}\), we consider deletion and relabeling. For \(h\), let \(\mathcal{H}^*(h) \subseteq \mathcal{H}\) be the set of edited decision trees \(h^*\) from \(h\) that satisfy the following five conditions:

**C1.** There exists an edit sequence \(e(h, h^*)\) consisting of deletion and relabeling operations to transform \(h\) into \(h^*\).

**C2.** All internal nodes \(i\) and leaves \(l\) of \(h^*\) have a branching feature \(d_i \in [D]\) and predictive label \(\hat{y}_i \in \mathcal{Y}\) as their node labels, respectively.

**C3.** All internal nodes \(i\) of \(h^*\) have two child nodes each: \(\text{left}(i)\) and \(\text{right}(i)\).

**C4.** For any leaf \(l\) of \(h^*\), there are no duplicate branching features for the internal nodes in \(A_l\).

**C5.** For any internal node \(i\) of \(h^*\), not all leaves included in its subtree have the same predictive label.

**C2** and **C3** serve to maintain the model structure of an edited decision tree \(h^*\), while **C4** and **C5** prevent \(h^*\) from including redundant nodes.

### 3.3 Problem Definition

Our goal is to reduce the discrimination score \(\Delta\) of a given decision tree \(h\) by editing its branching features and predictive labels without significantly changing it. Using \(\Gamma\) and \(\mathcal{H}^*(h)\), this problem can be defined as follows.

**Problem 1 (FADE problem)** Given a sample \(S\), a decision tree \(h\), a dissimilarity measure \(\Gamma\), and a discrimination threshold \(\theta \in [0, 1]\), find a decision tree \(h^*\) that is an optimal solution of the following optimization problem:

\[
\text{minimize } \Gamma(h, h^*) \quad \text{subject to } \Delta(h^* | S) \leq \theta.
\]

By solving the above optimization problem, we obtain an edited decision tree \(h^*\) that satisfies the fairness constraint \(\Delta(h^* | S) \leq \theta\) with minimum change with respect to the dissimilarity measure \(\Gamma(h, h^*)\).

### 4. MILO Formulation

In this chapter, we propose an MILO formulation of our FADE problem*1.

#### 4.1 Decision Tree Constraints

To express the edit operations for each node, i.e., deletion and relabeling (C1), and the edited decision tree \(h^* \in \mathcal{H}^*(h)\), we first introduce some program variables. We introduce variables \(\zeta_t \in \{0, 1\}\) for \(t \in \mathcal{T}\), which indicate whether a node \(t\) is deleted (\(\zeta_t = 0\)) or not (\(\zeta_t = 1\)). For an internal node \(i \in \mathcal{I}\) and leaf \(l \in \mathcal{L}\), let \(d^*_i \in [D]\) and \(\hat{y}^*_l \in \mathcal{Y}\) be their node labels in \(h^*\), respectively. For \(i \in \mathcal{I}\), we introduce variables \(\alpha_{i,d} \in \{0, 1\}\) for \(d \in [D]\), which indicate whether \(d^*_i = d\). Similarly, for \(l \in \mathcal{L}\), we introduce a variable \(\lambda^+_l \in \{0, 1\}\) (resp. \(\lambda^-_l \in \{0, 1\}\)) which indicates whether \(\hat{y}^*_l = 1\) (resp. \(\hat{y}^*_l = 0\)). We also introduce a variable \(\lambda^*_i \in \{0, 1\}\) (resp. \(\lambda^-_i \in \{0, 1\}\)) for \(i \in \mathcal{I}\), which indicates whether all leaves included in the subtree of \(i\) have the predictive label 1 (resp. 0). To satisfy the conditions mentioned in Section 3.2, we impose the following constraints on these variables:

\[
\begin{align*}
\sum_{d=1}^D \alpha_{i,d} &= \zeta_i, \forall i \in \mathcal{I}, \\
\lambda^+_l + \lambda^-_l &= \zeta_l, \forall l \in \mathcal{L}, \\
\zeta_i &\leq \zeta_j, \forall i, j \in \mathcal{I}, j_i = \max(A^R_i \cup \{0\}), \\
\zeta_i &= \zeta_l, \forall l \in \mathcal{L}, i_l = \max(A^R_i \cup \{0\}), \\
\sum_{i \in A_l} \alpha_{i,d} &\leq 1, \forall l \in \mathcal{L}, d \in [D], \\
0 &\leq \zeta_{\text{right}(i)} + 2 \cdot \lambda^*_i - \lambda^*_{\text{left}(i)} - \lambda^-_{\text{right}(i)} \leq 1, \forall i \in \mathcal{I}, s \in \{+, -\}, \\
\zeta_i &\leq 1 - (\lambda^+_i + \lambda^-_i), \forall i \in \mathcal{I}, \\
\zeta_0 &= 1.
\end{align*}
\]

*1 For the details of techniques of MILO formulation in this paper, see, e.g., [Williams 13].
where left \((i) \in T\) (resp. right \((i) \in T\)) is a left (resp. right) child node of \(i\), and \(\zeta_i\) is a dummy variable for notational convenience. Constraints (1–4) represent \(C2\) and \(C3\), while constraints (5–7) represent \(C4\) and \(C5\). Note that \(j_i\), left \((i)\), right \((i)\) for \(i \in I\) and \(i_t\) for \(l \in L\) can be computed when \(h\) is given.

### 4.2 Fairness Constraints

To express the fairness constraint \(\Delta(h^* | S) \leq \theta\), we must first express the output \(h^*(x(n))\) of an edited decision tree \(h^*\) for each input instance \(x(n)\) in a given sample \(S\). We introduce indicator variables \(\phi_{i,n} \in \{0, 1\}\) for \(l \in L\) and \(n \in [N]\) such that \(\phi_{i,n} = \phi^*_l(x(n))\), where \(\phi^*_l : X \to \{0, 1\}\) is the leaf indicator function of a leaf \(l\) in \(h^*\). We also introduce auxiliary variables \(\psi_{i,n} \in \{0, 1\}\) indicating whether \(x(n)\) satisfies the branching rule of \(i \in I\), i.e.,

\[\psi_{i,n} = \mathbb{I}[x_{d_l} = 1].\]

From the definition of the leaf indicator function in Section 2.1, we can express \(\phi_{i,n}\) as

\[\phi_{i,n} = \prod_{i \in A^L} (1 - \psi_{i,n}) \cdot \prod_{i \in A^R} \psi_{i,n}.
\]

These non-linear constraints can be linearized as follows:

1. \(\psi_{i,n} = \sum_{d=1}^{D} \alpha_{i,d} \cdot x_{d}^{(n)}, \forall i \in I, n \in [N],\)
2. \(I_t \cdot \phi_{i,n} \leq \sum_{i \in A^L} (1 - \psi_{i,n}) + \sum_{i \in A^R} \psi_{i,n}, \forall l \in L, n \in [N],\)
3. \(\sum_{i \in A^L} (1 - \psi_{i,n}) + \sum_{i \in A^R} \psi_{i,n} \leq I_t \cdot \phi_{i,n} + I_t - 1, \forall l \in L, n \in [N],\)

where \(I_t\) is a constant value such that \(I_t = |A^L|\), which can be computed when \(h\) is given. Constraint (9) represents \(\psi_{i,n} = \mathbb{I}[x_{d_l} = 1]\), while Constraints (10) and (11) represent \(\phi_{i,n} = \prod_{i \in A^L} (1 - \psi_{i,n}) \cdot \prod_{i \in A^R} \psi_{i,n}.\) For maintaining the model structure of \(h^*\) with respect to \(S\), the variables \(\phi_{i,n}\) must also satisfy the following constraints:

\[\zeta_l \leq \sum_{n \in [N]} \phi_{i,n}, \forall l \in L,\]
\[\phi_{i,n} \leq \zeta_l, \forall l \in L, n \in [N],\]
\[\sum_{i \in L} \phi_{i,n} = 1, \forall n \in [N].\]

Constraints (12) and (13) are the constraints for preventing \(h^*\) from having redundant leaves that no input instance in \(S\) reaches. Constraint (14) imposes that each input instance is assigned to exactly one leaf of \(h^*\).

Here, we express the fairness constraint using \(\phi_{i,n}\), and show that \(\Delta(h^* | S)\) can be expressed as the summation of the values of the DP score in each leaf of \(h^*\):

\[\Delta(h^* | S) = \sum_{i \in L} \lambda^+_l \cdot \sum_{n=1}^{N} c_n \cdot \phi_{i,n},\]

where \(c_n = \frac{x(n)}{N} - \frac{1}{N}.\)

\[\text{(Proposition 1)}\]

For \(\varepsilon \in \{0, 1\}\) and \(S\), let \(N_\varepsilon = \{n \in [N] | z(n) = \varepsilon\}\). Then, we have

\[\Delta(h^* | S) = \sum_{i \in L} \lambda^+_l \sum_{n=1}^{N} c_n \cdot \phi_{i,n},\]

\[\text{where } c_n = \frac{x(n)}{N} - \frac{1}{N}.\]

\[\text{\{Proof\}}\]

From the definitions of DP \(\Delta\) and decision tree \(h^*\), we have

\[\Delta(h^* | S) = \sum_{i \in L} \lambda^+_l \cdot \sum_{n=1}^{N} c_n \cdot \phi_{i,n}.
\]

We introduce variables \(\delta_l \in [-1, 1]\) for \(l \in L\) such that \(\delta_l = \lambda^+_l \cdot \sum_{n=1}^{N} c_n \cdot \phi_{i,n}\), which contain non-linear constraints on the variables \(\delta_l\), \(\lambda^+_l\), and \(\phi_{i,n}\). From Constraint 2 and \(-1 \leq \sum_{n=1}^{N} c_n \cdot \phi_{i,n} \leq 1\), we can linearize them by the following linear constraints:

\[\lambda^+_l \leq \delta_l \leq \lambda^+_l, \forall l \in L,
\]

\[\sum_{n=1}^{N} c_n \cdot \phi_{i,n} - \lambda^-_l \leq \delta_l \leq \sum_{n=1}^{N} c_n \cdot \phi_{i,n} + \lambda^-_l, \forall l \in L.
\]

Note that \(c_n\) is a constant value because it can be computed when \(S\) is given. From Proposition 1, the constraint \(\Delta(h^* | S) \leq \theta\) is equivalent to \(\sum_{l \in L} \delta_l \leq \theta\). It can be expressed as the following linear constraint of \(\delta_l\):

\[\sum_{l \in L} \delta_l \leq \theta.
\]

### 4.3 Objective Function

We introduce a variable \(\gamma \geq 0\) to express \(\gamma = \Gamma(h, h^*)\), which is the value of the objective function of Problem 1 to be minimized. In the following, we express PD and ED by linear constraints on program variables.

\[\text{§ 1 Prediction Discrepancy}\]

As with \(\Delta(h^* | S)\), we show that PD \(\Gamma_{PD}(h, h^* | S)\) can be expressed as the summation of the PD values in each leaf of \(h^*\) as follows:

\[\Gamma_{PD}(h, h^* | S) = \frac{1}{N} \sum_{n=1}^{N} \sum_{i \in L} \left(h(x(n)) - \hat{y}_i \cdot \phi_{i}^{(n)}\right)\]

\[= \frac{1}{N} \sum_{i \in L} \left(1 - \hat{y}_i \cdot \sum_{n=1}^{N} h(x(n)) \cdot \phi_{i}^{(n)} + \hat{y}_i \cdot \sum_{n=1}^{N} (1 - h(x(n))) \cdot \phi_{i}^{(n)}\right)
\]

\[= \frac{1}{N} \sum_{i \in L} (\lambda^-_l \cdot N^-_l + \lambda^+_l \cdot N^+_l),\]
where $N_l^+ = \sum_{n=1}^{N} h(x^{(n)}) \cdot \phi_l(x^{(n)})$ and $N_l^- = \sum_{n=1}^{N} (1 - h(x^{(n)})) \cdot \phi_l(x^{(n)})$ for $l \in L$.

Based on the above result, we introduce variables $\gamma_l \geq 0$ for $l \in L$ such that $\gamma_l = N_l^+ \cdot \lambda_l^- + N_l^- \cdot \lambda_l^+$, which is the value of PD in $l$. Since the values of $N_l^+$ and $N_l^-$ depend on the value of $h(x^{(n)})$, they include non-linear terms of the variables $\phi_{l,n}$, $\lambda_l^+$, and $\lambda_l^-$. We can linearize these non-linear constraints as follows:

$$N_l^- - N \cdot \lambda_l^- \leq \gamma_l \leq N_l^+ - N \cdot \lambda_l^+, \forall l \in L,$$

(18)

$$N_l^+ - N \cdot \lambda_l^+ \leq \gamma_l \leq N_l^- + N \cdot \lambda_l^-, \forall l \in L,$$

(19)

$$N_l^+ = \sum_{n=1}^{N} h(x^{(n)}) \cdot \phi_{l,n}, \forall l \in L,$$

(20)

$$N_l^- = \sum_{n=1}^{N} (1 - h(x^{(n)})) \cdot \phi_{l,n}, \forall l \in L.$$  

(21)

Note that $h(x^{(n)})$ is a constant value because they can be computed when $h$ and $S$ are given. Finally, we can express $\gamma = \Gamma_{PD}(h, h^* | S)$ by the following constraints:

$$\gamma = \frac{1}{N} \sum_{l \in \Gamma} \gamma_l.$$  

(22)

§ 2 Edit Distance

To formulate the ED $\Gamma_{ED}(h, h^*)$, we extend the formulation proposed in [Kondo 14] to address our FADE problem. [Kondo 14] proposed an integer linear optimization formulation for computing ED between given two trees by utilizing Tai mappings [Tai 79]. A Tai mapping $M$ between two labeled ordered trees $T_1$ and $T_2$ is a subset of $T_1 \times T_2$ satisfying the following three conditions for any $(t_1, t_2), (u_1, u_2) \in M$:

C6. (One-to-one mapping) $t_1 = u_1 \iff t_2 = u_2$,

C7. (Preserving ancestor) $t \prec_{\text{anc}} u_1 \iff t \prec_{\text{anc}} u_2$,

C8. (Preserving sibling) $t \prec_{\text{sib}} u_1 \iff t \prec_{\text{sib}} u_2$,

where $t \prec_{\text{anc}} u$ (resp. $t \prec_{\text{sib}} u$) indicates that a node $t$ is an ancestor (resp. to the left) of a node $u$. For a Tai mapping $M$, the cost of $M$ is defined as

$$\text{cost}(M) = \sum_{(t_1, t_2) \in M} \text{dist}(t_1, t_2) + \sum_{t_1 \in M_{\text{del}}} \text{dist}(t_1, \epsilon) + \sum_{t_2 \in M_{\text{ins}}} \text{dist}(\epsilon, t_2),$$

where $M_{\text{del}} = \{t_1 \in T_1 | \forall t_2 \in T_2 : (t_1, t_2) \notin M\}$ and $M_{\text{ins}} = \{t_2 \in T_2 | \forall t_1 \in T_1 : (t_1, t_2) \notin M\}$, which are the sets of deleted and inserted nodes, respectively. [Tai 79] proved that the ED between $T_1$ and $T_2$ is equivalent to the minimum Tai mapping cost, which indicates that the ED can be computed by finding a minimum-cost Tai mapping instead of finding a minimum-cost sequence of edit operations transforming $T_1$ into $T_2$. Based on this result, we formulate $\Gamma_{ED}(h, h^*)$ by expressing $\text{cost}(M)$ by program variables and minimizing it.

To express a Tai mapping $M$ between $h$ and $h^*$, we first introduce variables $\mu_{t_1, t_2} \in \{0, 1\}$ for $t_1, t_2 \in T$ that indicate whether $(t_1, t_2) \in M$. Since $M$ is a Tai mapping, $\mu_{t_1, t_2}$ must satisfy the following constraints [Kondo 14]:

$$\mu_{t_1, t_2} \in \{0, 1\}, \forall t_1, t_2 \in T,$$

(23)

$$\sum_{t_2 \in T} \mu_{t_1, t_2} \leq 1, \forall t_1 \in T,$$

(24)

$$\sum_{t_1 \in T} \mu_{t_1, t_2} = 0, \forall t_2 \in T,$$

(25)

$$\mu_{t_1, t_2} + \mu_{u_1, u_2} \leq 1, \forall ((t_1, t_2), (u_1, u_2)) \in T_{\text{anc}},$$

(26)

$$\mu_{t_1, t_2} + \mu_{u_1, u_2} \leq 1, \forall ((t_1, t_2), (u_1, u_2)) \in T_{\text{sib}},$$

(27)

where $T_{\text{anc}} = \{((t_1, t_2), (u_1, u_2)) \in T^2 \times T^2 | t_1 < u_1 \iff t_2 < u_2\}$ for a binary relation $< \in \{\text{anc}, \text{sib}\}$ on $T$, which can be computed when $h$ is given. Constraints (24) and (25) represents C6 (one-to-one mapping), while Constraint (26) and Constraint (27) represents C7 (preserving ancestor) and C8 (preserving sibling), respectively.

Next, we show that $\text{cost}(M)$ can be expressed as the summation of the costs in each node. In our Problem 1, since we consider deletion and relabeling as our available edit operation (C1), $M_{\text{del}} = \emptyset$ and $\sum_{t_2 \in M_{\text{ins}}} \text{dist}(\epsilon, t_2) = 0$ hold. From the definitions of $M_{\text{del}}$ and $\mu_{t_1, t_2}$, we obtain

$$\text{cost}(M) = \sum_{(t_1, t_2) \in M} \text{dist}(t_1, t_2)$$

$$+ \sum_{t_1 \in M_{\text{del}}} \text{dist}(t_1, \epsilon) + \sum_{t_2 \in M_{\text{ins}}} \text{dist}(\epsilon, t_2).$$

Note that $\text{dist}(t_1, \epsilon) = 1$ from the definition of dist.

We introduce variables $\gamma_{t_1} \geq 0$ for $t_1 \in T$ such that $\gamma_{t_1} = 1 + \sum_{t_2 \in T} (\text{dist}(t_1, t_2) - 1) \cdot \mu_{t_1, t_2}$. They include non-linear constraints because the value of dist($t_1, t_2$) depends on the node labels of $t_2$, i.e., the variables $\alpha_{t_2,d}$ if $t_2 \in T$, and $\lambda_{t_2}^+$ and $\lambda_{t_2}^-$ if $t_2 \notin T$. From the definition of $\mu_{t_1, t_2}$ and Constraint (25), we can express $\gamma_{t_1}$ as follows:

$$\gamma_{t_1} = \begin{cases} 1 & \text{if } \sum_{t_2 \in T} \mu_{t_1, t_2} = 0, \\
\text{dist}(t_1, t^*) & \text{otherwise,} \end{cases}$$

where $t^* \in T$ such that satisfies $\mu_{t_1, t^*} = 1$. From these observations, we can linearize the above constraints on $\gamma_{t_1}$.
by the following constraints:

\[ \gamma_t \geq 1 - \sum_{t_2 \in T} \mu_{t_1,t_2}, \forall t_1 \in T \]
\[ \gamma_t \geq \text{dist}(t_1,t_2) - U \cdot (1 - \mu_{t_1,t_2}), \forall t_1,t_2 \in T \]
\[ \text{dist}(t_1,t_2) = \begin{cases} \hat{y}_{t_1} \cdot \lambda_{t_2} + (1 - \hat{y}_{t_1}) \cdot \lambda_{t_2}^\perp & \text{if } t_1,t_2 \in L, \\ \sum_{d \in [D]} I[d \neq d'], \alpha_{t_1,d} & \text{if } t_1,t_2 \in T, \\ 1 & \text{otherwise,} \end{cases} \]

(28)  
(29)  
(30)

where \( U \geq 0 \) is the upper bound of \( \text{dist}(t_1,t_2) \), and we can set \( U = 1 \) from the definition of \( \text{dist} \). Finally, we can express \( \gamma = \Gamma_{\text{ED}}(h,h^*) \) by the following constraints:

\[ \gamma = \sum_{t \in T} \gamma_t. \]  
(31)

### 4.4 Overall Formulation

Finally, our overall MILO formulation can be expressed as follows:

\[
\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad \left\{ \begin{array}{l}
\text{Constraint (1–17),} \\
\text{Constraint (18–22), if } \Gamma = \Gamma_{\text{PD}}, \\
\text{Constraint (23–31), if } \Gamma = \Gamma_{\text{ED}},
\end{array} \right. \\
& \quad \alpha_{t,d} \in \{0,1\}, \forall t \in T, d \in [D], \\
& \quad \zeta_t, \lambda_t^+, \lambda_t^\perp \in \{0,1\}, \gamma_0 \geq 0, \forall t \in T, \\
& \quad \phi_{t,n} \in \{0,1\}, \delta_t \in [-1,1], \forall l \in L, n \in [N], \\
& \quad \gamma \geq 0.
\end{align*}
\]

The total numbers of variables and constraints excluding those for \( \Gamma \) are both \( O(|L| \cdot (N + D)) \). For \( \Gamma = \Gamma_{\text{PD}} \) (resp. \( \Gamma = \Gamma_{\text{ED}} \), \( O(|L|) \) variables and constraints (resp. \( O(|T|^4) \) variables and \( O(|T|^2) \) constraints) are additionally required.

As with existing MILO-based methods [Aghaei 19, Bertsimas 17, Verwer 19], our formulation can be efficiently solved by off-the-shelf MILO solvers, such as CPLEX [IBM 20], and customized by adding constraints that users desire. Consequently, we can obtain an optimal edited decision tree under not only fairness constraints but also user-defined additional constraints without implementing designated algorithms.

### Additional Constraints

Our FADE problem includes no constraints on the prediction error of an edited decision tree \( h^* \). To improve the practicality of the obtained decision tree, we can add constraints corresponding to the empirical risk to our formulation. We denote the empirical risk of a classifier \( h \) on a sample \( S \) by \( R(h \mid S) = \frac{1}{N} \sum_{(x,y,z) \in S} I[y \neq h(x)] \). For a parameter \( \varepsilon \geq 0 \), we consider the constraint \( R(h^* \mid S) \leq (1 + \varepsilon) \cdot R(h \mid S) \), which states that the increase in the empirical risk of an edited decision tree \( h^* \) must be within \( 100\varepsilon \% \) of the initial decision tree \( h \). This constraint can be expressed as follows:

\[ \rho_t \geq 0, \forall l \in L \]
\[ R_t^+ - N \cdot \lambda_t^+ \leq \rho_t \leq R_t^+ + N \cdot \lambda_t^\perp, \forall l \in L, \ast \in \{+, -, \} \]
\[ R_t^+ = \sum_{n=1}^{N} y^{(n)} \cdot \phi_{t,n}, \forall l \in L, \\
\frac{1}{N} \sum_{l \in L} \rho_l \leq (1 + \varepsilon) \cdot R(h \mid S). \]

These constraints are derived by replacing \( h(x^{(n)}) \) and \( \gamma_l \) in Constraints (18–21), which are the constraints for expressing PD \( \Gamma_{\text{PD}} \), with \( y^{(n)} \) and \( \rho_t \), respectively. In our experiments, we added the above constraints into our overall formulation.

### 5. Experiments

In this chapter, we conducted experiments on real datasets to evaluate our proposed FADE method. All code was implemented in Python 3.7 with scikit-learn and IBM ILOG CPLEX v12.10, and all experiments were conducted on 64-bit macOS Catalina 10.15.6 with Intel Core i9 2.4 GHz CPU and 64 GB memory. In addition, a 5-hour time limit was imposed for obtaining the solution.

#### 5.1 Experimental Setup

We consider out-of-sample deployment settings [Ustun 19b], where a classifier is trained on a biased dataset and is deployed on individuals who are not included in the training dataset. We investigate the behavior and effectiveness of FADE in such scenarios. To reproduce this setting and apply FADE to it, we performed the following procedures:

1. Divide a training sample \( S \) into \( S_0 \) and \( S_1 \), where \( S_z = \{ (x,y,z) \in S \mid z = z \} \) for \( z \in \{0,1\} \).
2. Train a decision tree \( h \) on the biased sample \( S_0 \).
3. Edit \( h \) on the entire training sample \( S \) by FADE.

In our preliminary experiments, we confirmed that procedures (1) and (2) often increased the DP values of the initial decision trees \( h \). We trained the initial decision trees \( h \) by CART [Breiman 84] with a maximum depth of 4. We denote the method that optimizes \( \Gamma_{\text{PD}} \) (resp. \( \Gamma_{\text{ED}} \)) as the dissimilarity measure \( \Gamma \) by \( \text{FADE-PD} \) (resp. \( \text{FADE-ED} \)).

#### Datasets

We used the German (\( N = 1000, D = 31 \)) [Dua 17] and COMPAS (\( N = 6167, D = 17 \)) [Larson 16] datasets, which are standard benchmarking datasets in the context of fairness in machine learning [Hajian 16]. The task of the German dataset is to predict whether individuals will default on their loan, while the task of the COMPAS dataset...
Performance Comparison with Baseline

\[ \Gamma(h, h^*) \]

became more difficult to edit a biased decision tree \( h \) corresponding to a discrimination score \( \Delta \) decreased. This indicates that as the constraint corresponding to a discrimination score \( \Delta \) became stronger, it became more difficult to edit a biased decision tree \( h \) into a fair decision tree \( h^* \) with fewer changes with respect to the dissimilarity \( \Gamma(h, h^*) \).

5.2 Experimental Results

§ 1 Sensitivity Analysis of Discrimination Threshold

To evaluate the behavior of our proposed framework, we show the sensitivity of the threshold \( \theta \) for the discrimination score \( \Delta(h^* | S) \) in our optimization problem. Figure 3 presents \( \Gamma(h, h^*) \) between an initial decision tree \( h \) and its edited one \( h^* \) for each \( \theta \). It can be seen that both PD and ED between \( h \) and \( h^* \) monotonically increased as \( \theta \) decreased. This indicates that as the constraint corresponding to a discrimination score \( \Delta \) became stronger, it became more difficult to edit a biased decision tree \( h \) into a fair decision tree \( h^* \) with fewer changes with respect to the dissimilarity \( \Gamma(h, h^*) \).

§ 2 Performance Comparison with Baseline

Next, we compared the performance of decision trees edited by FADE with Baseline. We set \( \theta = 0.01, \varepsilon = 0.2 \) for the German dataset and \( \theta = 0.1, \varepsilon = 0.3 \) for the COMPAS dataset. We measured the average empirical risk \( R \) and DP \( \Delta \). We also report the average values of PD \( \Gamma_{PD} \) and ED \( \Gamma_{ED} \) between the initial biased decision tree trained by CART and decision trees obtained by each method.

Table 1 presents the results of 10-fold cross validation. Although our FADE edited biased decision trees, their empirical risk and DP were only approximately 3% lower than those obtained by Baseline. In addition, we can also see that FADE achieved lower PD and ED than those obtained by Baseline. For example, in the German dataset, the test risk, test DP, PD, and ED given by FADE-ED were 0.298, 0.066, 0.172, and 1.0, respectively, which were comparable to or less than those obtained by Baseline. These results indicate that our proposed method FADE can convert biased decision trees into fair decision tree without significantly changing them.

Computational Complexity. Whereas FADE-ED often succeeded in finding an optimal solution within the predefined time limit, FADE-PD did not for either the German or COMPAS datasets. Actually, the average running time of FADE-ED was 187 seconds on the German dataset and 4786 seconds on the COMPAS dataset, while that of FADE-PD was 18000 seconds on both the German and COMPAS datasets since FADE-PD never succeeded in finding optimal solutions within 5-hours. These results seem counterintuitive because the total number of variables and constraints included in FADE-PD is less than that in FADE-ED as shown in Section 4.4. These results indicate that minimizing PD \( \Gamma_{PD} \) under the fairness constraint with respect to DP \( \Delta \) is more computationally difficult than ED \( \Gamma_{ED} \).

Edited Rules. Finally, we present examples of branching features edited by FADE in Table 2, which displays pairs of replaced branching rules to eliminate discrimination from the initial decision trees. These results can help decision-makers to identify the causes of unfairness in their deployed models, and suggest ways to eliminate them [Adler 18, Pedreshi 08].

6. Conclusion

In this study, we proposed a fairness-aware post-processing method for converting a trained decision tree into a fair decision tree without significantly changing it in terms of its prediction and interpretation. For this purpose, we introduced two dissimilarity measures between decision trees based on the prediction discrepancy and edit distance, and proposed a mixed-integer linear optimization approach for
optimize the decision tree by considering fairness constraints. Through experiments on real datasets, we confirmed the effectiveness of our method by comparing it with an existing in-processing method for learning fair decision trees.

In future work, we plan to devise a more efficient MILO formulation. Moreover, it would be interesting to develop a more designed dissimilarity measure between decision trees by considering the depth of nodes in the decision trees and the dissimilarity between the node labels. It is also interesting future work to extend our framework to the online setting [Zhang 19].

Acknowledgments

We thank anonymous reviewers for their insightful comments. This work was supported by JSPS KAKENHI Grant-in-Aid for JSPS Research Fellow 20J20654, Scientific Research (A) 20H00595, and JST CREST JPMJCR18K3.

Table 1: Results of 10-fold cross validation.

(a) German dataset

| Method     | Empirical Risk |       |       | DP   |       |       | PD | ED |
|------------|----------------|-------|-------|------|-------|-------|----|----|
|            | Train          | Test  | Train | Test | Train | Test  |    |    |
| Baseline   | 0.269 ± 0.0082 | 0.295 | 0.201 |      | 0.00599 ± 0.0041 | 0.0816 ± 0.049 | 0.251 ± 0.035 | 16.2 ± 2.1 |
| FADE-PD    | 0.289 ± 0.012  | 0.318 | 0.047 |      | 0.00449 ± 0.0031 | 0.0951 ± 0.09  | 0.0907 ± 0.018 | 4.9 ± 2.6 |
| FADE-ED    | 0.288 ± 0.014  | 0.298 | 0.029 |      | 0.00371 ± 0.0025 | 0.066 ± 0.05   | 0.172 ± 0.051 | 1.0 ± 0.0 |

(b) COMPAS dataset

| Method     | Empirical Risk |       |       | DP   |       |       | PD | ED |
|------------|----------------|-------|-------|------|-------|-------|----|----|
|            | Train          | Test  | Train | Test | Train | Test  |    |    |
| Baseline   | 0.395 ± 0.014  | 0.399 | 0.027 |      | 0.0679 ± 0.026 | 0.0786 ± 0.03  | 0.325 ± 0.041 | 15.1 ± 2.2 |
| FADE-PD    | 0.402 ± 0.0071 | 0.401 | 0.022 |      | 0.092 ± 0.0081 | 0.0953 ± 0.03  | 0.204 ± 0.045 | 8.2 ± 3.4 |
| FADE-ED    | 0.414 ± 0.0088 | 0.413 | 0.014 |      | 0.0757 ± 0.022 | 0.0718 ± 0.032 | 0.268 ± 0.038 | 1.7 ± 0.67 |

Table 2: Examples of edited branching features by FADE.

(a) German dataset

| Initial Rule | Edited Rule |
|--------------|-------------|
| Job Skill ≤ 2 | Duration ≤ 10 |
| Duration ≤ 10 | Credit Amount ≤ 3368 |
| Duration ≤ 10 | Checking Account ≤ 2 |
| Checking Account ≤ 1 | Housing = Rent |

(b) COMPAS dataset

| Initial Rule | Edited Rule |
|--------------|-------------|
| Priors ≤ 1 | Charge Degree = Felonies |
| Priors ≤ 3 | Juvenile-Misdemeanors ≤ 0 |
| Age ≤ 31 | Juvenile-Felonies ≤ 0 |
| Juvenile-Felonies ≤ 0 | Age ≤ 27 |

References

[Adler 18] Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S.: Auditing Black-Box Models for Indirect Influence, Knowledge and Information Systems, Vol. 54, No. 1, pp. 95–122 (2018)

[Aghaei 19] Aghaei, S., Azizi, M. J., and Vayanos, P.: Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making, in Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 1418–1426 (2019)

[Aglin 20] Aglin, G., Nijsen, S., and Schaus, P.: Learning Optimal Decision Trees Using Caching Branch-and-Bound Search, in Proceedings of the 34th AAAI Conference on Artificial Intelligence, pp. 3146–3153 (2020)

[Aivodji 19] Aivodji, U., Arai, H., Fortineau, O., Gambs, S., Hara, S., and Tapp, A.: Fairwashing: The Risk of Rationalization, in Proceedings of the 36th International Conference on Machine Learning, pp. 161–170 (2019)

[Barocas 16] Barocas, S. and Selbst, A. D.: Big Data’s Disparate Impact, California Law Review, Vol. 104, No. 3, pp. 671–732 (2016)

[Bertsimas 17] Bertsimas, D. and Dunn, J.: Optimal Classification Trees, Machine Learning, Vol. 106, No. 7, pp. 1039–1082 (2017)

[Breiman 84] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J.: Classification and Regression Trees, V ol. 161–170 (2019)

[Bertsimas 17] Bertsimas, D. and Dunn, J.: Optimal Classification Trees, Machine Learning, V ol. 121, No. 2, pp. 277–292 (2010)

[Calmon 17] Calmon, F. P., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R.: Optimized Pre-Processing for Discrimination Prevention, in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 3995–4004 (2017)

[Carreira-Perpiñán 18] Carreira-Perpiñán, M. A. and Tavallali, P.: Alternating Optimization of Decision Trees, with Application to Learning Sparse Oblique Trees, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 1219–1229 (2018)

[Dua 17] Dua, D. and Graff, C.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml (2017), Accessed Jan. 15th, 2021

[Guidotti 19] Guidotti, R. and Ruggieri, S.: On The Stability of Interpretable Models, in Proceedings of 2019 International Joint Confer-
ence on Neural Networks, pp. 1–8 (2019)

[Hajian 16] Hajian, S., Bonchi, F., and Castillo, C.: Algorithmic Bias: From Discrimination Discovery to Fairness-Aware Data Mining, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2125–2126 (2016)

[Hardt 16] Hardt, M., Price, E., and Srebro, N.: Equality of Opportunity in Supervised Learning, in Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 3323–3331 (2016)

[Hu 19] Hu, X., Rudin, C., and Seltzer, M.: Optimal Sparse Decision Trees, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 7265–7273 (2019)

[IBM 20] IBM: CPLEX Optimizer — IBM, https://www.ibm.com/analytics/cplex-optimizer (2020), Accessed Jan. 15th, 2021

[Kamiran 10] Kamiran, F., Calders, T., and Pechenizkiy, M.: Discrimination-Aware Decision Tree Learning, in Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 869–874 (2010)

[Kamiran 12] Kamiran, F. and Calders, T.: Data Preprocessing Techniques for Classification without Discrimination, Knowledge and Information Systems, Vol. 33, No. 1, pp. 1–33 (2012)

[Kamishima 12] Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J.: Fairness-Aware Classifier with Prejudice Remover Regularizer, in Proceedings of the 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 35–50 (2012)

[Kondo 14] Kondo, S., Otaki, K., Ieda, M., and Yamamoto, A.: Fast Computation of the Tree Edit Distance between Unordered Trees Using IP Solvers, in Proceedings of the 17th International Conference on Discovery Science, pp. 156–167 (2014)

[Larson 16] Larson, J., Mattu, S., Kirchner, L., and Angwin, J.: How We Analyzed the COMPAS Recidivism Algorithm, https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm (2016), Accessed Jan. 15th, 2021

[Marx 20] Marx, C., Calmon, F., and Ustun, B.: Predictive Multiplicity in Classification, in Proceedings of the 37th International Conference on Machine Learning, pp. 6765–6774 (2020)

[Pedreshi 08] Pedreshi, D., Ruggieri, S., and Turini, F.: Discrimination-Aware Data Mining, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 560–568 (2008)

[Quiñ 09] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D.: Dataset Shift in Machine Learning, The MIT Press (2009)

[Tai 79] Tai, K.–C.: The Tree-to-Tree Correction Problem, Journal of the ACM, Vol. 26, No. 3, pp. 422–433 (1979)

[Ustun 19a] Ustun, B., Liu, Y., and Parkes, D.: Fairness without Harm: Decoupled Classifiers with Preference Guarantees, in Proceedings of the 36th International Conference on Machine Learning, pp. 6573–6582 (2019)

[Ustun 19b] Ustun, B., Spangher, A., and Liu, Y.: Actionable Recourse in Linear Classification, in Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 10–19 (2019)

[Verwer 19] Verwer, S. and Zhang, Y.: Learning Optimal Classification Trees Using a Binary Linear Program Formulation, in Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 1625–1632 (2019)

[Wang 19] Wang, H., Ustun, B., and Calmon, F.: Repairing without Retraining: Avoiding Disparate Impact with Counterfactual Distributions, in Proceedings of the 36th International Conference on Machine Learning, pp. 6618–6627 (2019)

[Williams 13] Williams, H. P.: Model Building in Mathematical Programming, John Wiley & Sons, Inc., 5th edition (2013)

[Zafar 17] Zafar, M. B., Valera, I., Rodriguez, M. G., Gummadi, K. P., and Weller, A.: From Parity to Preference-Based Notions of Fairness in Classification, in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 228–238 (2017)

[Zafar 19] Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P.: Fairness Constraints: A Flexible Approach for Fair Classification, Journal of Machine Learning Research, Vol. 20, No. 75, pp. 1–42 (2019)

[Zhang 89] Zhang, K. and Shasha, D.: Simple Fast Algorithms for the Computation of the Tree Edit Distance between Trees and Related Problems, SIAM Journal on Computing, Vol. 18, No. 6, pp. 1245–1262 (1989)

[Zhang 19] Zhang, W. and Ntoutsi, E.: FAHT: An Adaptive Fairness-Aware Decision Tree Classifier, in Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 1480–1486 (2019)

Received January 15, 2021.