Does Machine Bring in Extra Bias in Learning? Approximating Discrimination Within Models Quickly

Yijun Bian*
Department of Computer Science
University of Copenhagen
yibi@di.ku.dk

Yujie Luo*
Department of Mathematics
National University of Singapore
lyj96@nus.edu.sg

Ping Xu
Department of Electrical and Computer Engineering
The University of Texas Rio Grande Valley
ping.t.xu@utrgv.edu

Abstract

Discrimination mitigation within machine learning (ML) models is complicated because multiple factors may interweave with each other including hierarchically and historically. Yet few existing fairness measures can capture the discrimination level within ML models when dealing with multiple sensitive attributes. To bridge this gap, we propose a fairness measure based on distances between sets from a manifold perspective, named '*harmonic fairness measure via manifolds (HFM)*' with three optional versions, which can deal with a fine-grained discrimination evaluation for several sensitive attributes of binary/multiple values. To accelerate the computation of distances of sets, we further propose approximation algorithms for efficient bias evaluation. The empirical results demonstrate that our proposed fairness measure *HFM* is valid and the approximation algorithms are effective and efficient.

1 Introduction

As techniques of machine learning (ML) and deep learning (DL) are flourishing developed and ML/DL systems are widely deployed in real life nowadays, the concern about the underlying discrimination hidden in these models has grown, particularly in high-stakes domains such as healthcare, recruitment, and jurisdiction [26], where equity for all stakeholders is pivotal to prevent unjust outcomes, akin to a discriminatory Matthew effect. It is of significance to prevent ML models from perpetuating or exacerbating inappropriate human prejudices for not only model performance but also societal welfare. Effectively addressing and eliminating discrimination usually requires a comprehensive grasp of its occurrence, causes, and mechanisms. For instance, a case involving a person changing their gender for lower car insurance rates highlights the complexity of fairness in ML.

Although the impressive practical advancements of ML and DL thrive on abundant data, their trustworthiness and equity heavily hinge on data quality. In fact, one of the primary sources of unfairness identified in the existing literature is biases from the data, possibly collected from various sources such as device measurements and historically biassed human decisions [32]. Moreover, the challenge of data imbalance often looms in human-sensitive domains, amplifying concerns of discrimination and bias propagation in ML models. Then misinformed model training would amplify imbalances and biases in data, with wide-reaching societal implications. For example, optimising aggregated prediction errors can advantage privileged groups over marginalised ones. In addition,

*Equal contribution.

Mathematics of Modern Machine Learning Workshop at NeurIPS 2024.

missing data like instances or values may introduce disparities between the dataset and the target population, leading to biassed results as well. Therefore, in order to ensure fairness and mitigate biases, it is crucial to correctly cope with data imbalance and prevent ML models from perpetuating or even exacerbating inappropriate human prejudices.

To mitigate bias within ML models, the very first step is to promptly recognise its occurrence. However, promptly detecting discrimination fully, truly, and faithfully is not quite easy because of plenty of factors interweaving with each other. First, learning algorithms might yield unfair outcomes even with purely clean data due to proxy attributes for sensitive features or tendentious algorithmic objectives. For instance, the educational background of one person might be a proxy attribute for those born in families with a preference for boys. Second, the existence of multiple sensitive attributes and their twist with each other highlight the complexity of bias tackling, like one member from a marginalised group could become one of the majority concerning another factor, or vice versa. Third, dynamic changes and historical factors may need to be taken into account, as bias hidden in data, data imbalance, and present decisions may interweave, causing some interrelated impact and vicious circles. Despite many fairness measures that have been proposed to facilitate bias mitigation, existing measures can only focus on either group or individual fairness, rather than incorporating both together. Besides, most of them mainly focus on one or more sensitive attributes with binary values, yet few could handle bias appropriately when facing multiple sensitive attributes with even multiple values. Therefore, it motivates us to investigate a proper tool to deal with bias in such aforementioned scenarios.

In this paper, we investigate the possibility of assessing the discrimination level of ML models in the existence of several sensitive attributes with multiple values. To this end, we introduce a novel fairness measure from a manifold perspective, named '*harmonic fairness measure via manifolds (HFM)*', with three optional versions (that is, previous, maximum, and average *HFM*). However, the direct calculation of *HFM* lies on a core distance between two sets, which might be pretty costly. Therefore, we further propose two approximation algorithms that quickly estimate the distance between sets, named '*Approximation of distance between sets for one sensitive attribute (ApproxDist)*' and '*Approximation of extended distance between sets for several sensitive attributes (ExtendDist)*' respectively, in order to speed up the calculation and enlarge its practical applicable values. Furthermore, we also investigate their algorithmic properties under certain reasonable assumptions, in other words, how effective they could be in achieving the approximation goal.

Our contribution in this work is three-fold: (1) We propose a fairness measure named *HFM* that could reflect the discrimination level of classifiers even simultaneously facing several sensitive attributes with multiple values. Note that *HFM* has three optional versions, of which all are built upon a concept of distances between sets from the manifold perspective. (2) We propose two approximation algorithms (that is, *ApproxDist* and *ExtendDist*) that accelerate the estimation of distances between sets, to mitigate its disadvantage of costly direct calculation of *HFM*. (3) Comprehensive experiments are conducted to demonstrate the effectiveness of the proposed *HFM* and approximation algorithms.

2 Methodology

In this section, we formally study the measurement of fairness from a manifold perspective.

We use $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ to denote a dataset where the instances are iid. (independent and identically distributed), drawn from an feature-label space $\mathcal{X} \times \mathcal{Y}$ based on an unknown distribution. The feature/input space $\mathcal{X}$ is arbitrary, and the label/output space $\mathcal{Y} = \{1, 2, ..., n_c\} (n_c \geqslant 2)$ is finite, which could be binary or multi-class classification, depending on the number of labels (i.e., the value of n_c). Presuming that the considered dataset S is composed of the instances including sensitive attributes, the features of one instance including sensitive attributes $\boldsymbol{a} = [a_1, a_2, ..., a_{n_a}]^\intercal$ is represented as $\boldsymbol{x} \triangleq (\breve{\boldsymbol{x}}, \boldsymbol{a})$, where $n_a \geqslant 1$ is the number of sensitive attributes allowing multiple attributes and $a_i \in \mathbb{Z}_+ (1 \leqslant i \leqslant n_a)$ allows both binary and multiple values. A function $f \in \mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$ represents a hypothesis in a space of hypotheses $\mathcal{F}$, of which the prediction for one instance $\boldsymbol{x}$ is denoted by $f(\boldsymbol{x})$ or $\hat{y}$. Note that $i \in [n]$ is used to represent $i \in \{1, 2, ..., n\}$ for brevity.

2.1 Model fairness assessment from a manifold perspective

Given the dataset $S = \{(\breve{\boldsymbol{x}}_i, \boldsymbol{a}_i, y_i) | i \in [n]\}$ composed of instances including sensitive attributes, here we denote one instance by $\boldsymbol{x} = (\breve{\boldsymbol{x}}, \boldsymbol{a}) = [x_1, ..., x_{n_x}, a_1, ..., a_{n_a}]^\intercal$ for clarity, where n_a is the

number of sensitive/protected attributes and n_x is that of unprotected attributes in $\boldsymbol{x}$. In this paper, we introduce a new fairness measure in scenarios of one or more sensitive attributes from a manifold perspective, which works for both binary and multiple possible values. Inspired by the principle of individual fairness—similar treatment for similar individuals, if we view the instances (with the same sensitive attributes) as data points on certain manifolds, the manifold representing members from the marginalised/unprivileged group(s) is supposed to be as close as possible to that representing members from the privileged group. To measure the fairness with respect to the sensitive attribute, we proposed a fairness measure that is inspired by 'the distance of sets' introduced in mathematics.

Distance between sets for one sensitive attribute with binary values For a certain bi-valued sensitive attribute $a_i \in \mathcal{A}_i = \{0, 1\}$, S can be divided into two subsets $S_1 = \{(\boldsymbol{x}, y) \in S \mid a_i = 1\}$ and $\bar{S}_1 = S \setminus S_1 = \{(\boldsymbol{x}, y) \in S \mid a_i \neq 1\}$, where $a_i = 1$ means the corresponding instance is a member from the privileged group. Then given a specific distance metric $\mathbf{d}(\cdot, \cdot)$ (e.g., the standard Euclidean metric) on the feature space, the distance between these two subsets (that is, S_1 and $\bar{S}_1$) is defined by

$$\mathbf{D}(S_1, \bar{S}_1) \triangleq \max\Big\{ \max_{(\boldsymbol{x},y)\in S_1} \min_{(\boldsymbol{x}',y')\in \bar{S}_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, y), (\breve{\boldsymbol{x}}', y')\big), \max_{(\boldsymbol{x}',y')\in \bar{S}_1} \min_{(\boldsymbol{x},y)\in S_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, y), (\breve{\boldsymbol{x}}', y')\big) \Big\}, \quad (1)$$

and it is viewed as the distance between the manifolds of marginalised group(s) and that of the privileged group. Notice that this distance satisfies three basic properties: identity, symmetry, and triangle inequality. Analogously, for a trained classifier $f(\cdot)$, we can calculate

$$\mathbf{D}_f(S_1, \bar{S}_1) = \max\Big\{ \max_{(\boldsymbol{x},y)\in S_1} \min_{(\boldsymbol{x}',y')\in \bar{S}_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, \hat{y}), (\breve{\boldsymbol{x}}', \hat{y}')\big), \max_{(\boldsymbol{x}',y')\in \bar{S}_1} \min_{(\boldsymbol{x},y)\in S_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, \hat{y}), (\breve{\boldsymbol{x}}', \hat{y}')\big) \Big\}. \quad (2)$$

By recording the true label y and the prediction $\hat{y}$ as one denotation (say $\ddot{y}$) for simplification, we could rewrite (1) and (2) as

$$\mathbf{D}_{\cdot}(S_1, \bar{S}_1) \triangleq \max\Big\{ \max_{(\boldsymbol{x},y)\in S_1} \min_{(\boldsymbol{x}',y')\in \bar{S}_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, \ddot{y}), (\breve{\boldsymbol{x}}', \ddot{y}')\big), \max_{(\boldsymbol{x}',y')\in \bar{S}_1} \min_{(\boldsymbol{x},y)\in S_1} \mathbf{d}\big((\breve{\boldsymbol{x}}, \ddot{y}), (\breve{\boldsymbol{x}}', \ddot{y}')\big) \Big\}. \quad (3)$$

We will continue using the above notations in the subsequent context for simplification.

Distance between sets for one sensitive attribute with multiple values As for the scenarios where only one sensitive attribute exists, let $\boldsymbol{a} = [a_i]^\mathsf{T}$ be a single sensitive attribute, in other words, $n_a = 1$, $a_i \in \mathcal{A}_i = \{1, 2, ..., n_{a_i}\}$, $n_{a_i} \geqslant 3$, and $n_{a_i} \in \mathbb{Z}_+$. Then the original dataset S can be divided into a few disjoint sets according to the value of this attribute a_i, that is, $S_j = \{(\boldsymbol{x}, y) \in S \mid a_i = j\}, \forall j \in \mathcal{A}_i$. We can now extend (3) and introduce the following distance measures: (i) *maximal distance measure for one sensitive attribute*

$$\mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i) \triangleq \max_{1\leqslant j\leqslant n_{a_i}} \Big\{ \max_{(\boldsymbol{x},y)\in S_j} \min_{(\boldsymbol{x}',y')\in \bar{S}_j} \mathbf{d}\big((\breve{\boldsymbol{x}}, \ddot{y}), (\breve{\boldsymbol{x}}', \ddot{y}')\big) \Big\}, \quad (4)$$

and (ii) *average distance measure for one sensitive attribute*

$$\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S, a_i) \triangleq \frac{1}{n} \sum_{j=1}^{n_{a_i}} \sum_{(\boldsymbol{x},y)\in S_j} \min_{(\boldsymbol{x}',y')\in \bar{S}_j} \mathbf{d}\big((\breve{\boldsymbol{x}}, \ddot{y}), (\breve{\boldsymbol{x}}', \ddot{y}')\big), \quad (5)$$

where $\bar{S}_j = S \setminus S_j$. Notice that $\mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i) = \mathbf{D}_{\cdot}(S_1, \bar{S}_1)$ when $\mathcal{A}_i = \{0, 1\}$.

Distance between sets for multiple sensitive attributes with multiple values Now we discuss the general case, where we have several sensitive attributes $\boldsymbol{a} = [a_1, a_2, ..., a_{n_a}]^\mathsf{T}$ and each $a_i \in \mathcal{A}_i = \{1, 2, .., n_{a_i}\}$, where n_{a_i} is the number of values for this sensitive attribute a_i $(1 \leqslant i \leqslant n_a)$. We can now introduce the following generalised distance measures: (i) *maximal distance measure for sensitive attributes*

$$\mathbf{D}_{\cdot,\boldsymbol{a}}(S) \triangleq \max_{1\leqslant i\leqslant n_a} \mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i), \quad (6)$$

and (ii) *average distance measure for sensitive attributes*

$$\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S) \triangleq \frac{1}{n_a} \sum_{i=1}^{n_a} \mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S, a_i). \quad (7)$$

Remark. (1) It is easy to see that $\mathbf{D}_{\cdot,\boldsymbol{a}}(S) \geqslant \mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S)$. (2) Both $\mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i)$ and $\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S, a_i)$ measure the fairness regarding the sensitive attribute a_i. (3) As their names suggest, the maximal distance represents the largest possible disparity between instances with different sensitive attributes, while the average distance reflects the average disparity between instances with different sensitive attributes. The formal distance measures are more stringent, they are susceptible to data noise. In contrast, the latter type of distance measures are more resilient against the influence of data noise.

Algorithm 1 Acceleration sub-procedure, aka. `AcceleDist` $(\{(\breve{\boldsymbol{x}}_i, a_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n, \boldsymbol{w}; m_2)$

Input: Data points $\{(\breve{\boldsymbol{x}}_i, a_i)\}_{i=1}^n$, its corresponding value $\{\ddot{y}_i\}_{i=1}^n$, where $\ddot{y}_i$ could be its true label y_i or prediction $\hat{y}_i$ by the classifier $f(\cdot)$, a random vector $\boldsymbol{w}$ for projection, and a hyper-parameter m_2
Output: Approximation of $\mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i)$ and $n\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S, a_i)$
1: Project data points onto a one-dimensional space based on (9), in order to obtain $\{g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})\}_{i=1}^n$
2: Sort original data points based on $\{g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})\}_{i=1}^n$ as their corresponding values, in ascending order
3: **for** i from 1 to n **do**
4: Set the anchor data point $(\boldsymbol{x}_i, \ddot{y}_i)$ in this round
5: // `If` $a_i = j$ `(marked for clarity), to approximate` $\min_{(\boldsymbol{x}', y') \in \bar{S}_j} \mathbf{d}\big((\breve{\boldsymbol{x}}_i, \ddot{y}_i), (\breve{\boldsymbol{x}}', \ddot{y}')\big)$
6: Compute the distances $\mathbf{d}((\breve{\boldsymbol{x}}_i, \ddot{y}_i), \cdot)$ for at most m_2 nearby data points that meets $a \neq a_i$ and $g(\breve{\boldsymbol{x}}, \ddot{y}; \boldsymbol{w}) \leqslant g(\breve{\boldsymbol{x}}_i, \ddot{y}_i; \boldsymbol{w})$
7: Find the minimum among them, recorded as $d^s_{\min}$
8: Compute the distances $\mathbf{d}((\breve{\boldsymbol{x}}_i, \ddot{y}_i), \cdot)$ for at most m_2 nearby data points that meets $a \neq a_i$ and $g(\boldsymbol{x}, \ddot{y}; \boldsymbol{w}) \geqslant g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})$
9: Find the minimum among them, recorded as $d^r_{\min}$
10: $d^{(i)}_{\min} = \min\{d^s_{\min}, d^r_{\min}\}$
11: **end for**
12: **return** $\max\{d^{(i)}_{\min} \mid i \in [n]\}$ and $\sum_{i=1}^n d^{(i)}_{\min}$

We remark that $\mathbf{D}_{\boldsymbol{a}}(S)$, $\mathbf{D}^{\text{avg}}_{\boldsymbol{a}}(S)$ reflect the biases from the data and $\mathbf{D}_{f,\boldsymbol{a}}(S)$, $\mathbf{D}^{\text{avg}}_{f,\boldsymbol{a}}(S)$ reflect the extra biases from the learning algorithm. Then the following values could be used to reflect the fairness degree of this classifier, that is,

$$\mathbf{df}_{\text{prev}}(f) = {}^{\mathbf{D}_{f,\boldsymbol{a}}(S)}\!/_{\mathbf{D}_{\boldsymbol{a}}(S)} - 1\,, \tag{8a}$$

$$\mathbf{df}(f) = \log\left({}^{\mathbf{D}_{f,\boldsymbol{a}}(S)}\!/_{\mathbf{D}_{\boldsymbol{a}}(S)}\right)\,, \tag{8b}$$

$$\mathbf{df}^{\text{avg}}(f) = \log\left({}^{\mathbf{D}^{\text{avg}}_{f,\boldsymbol{a}}(S)}\!/_{\mathbf{D}^{\text{avg}}_{\boldsymbol{a}}(S)}\right)\,. \tag{8c}$$

We name the fairness degrees of one classifier, defined above by (8), as '*previous harmonic fairness measure via manifolds (HFM)*', '*maximum HFM*', and '*average HFM*', respectively.

2.2 A prompt approximation of distances between sets for Euclidean spaces

To reduce the high computational complexity ($\mathcal{O}(n^2)$) of directly calculating (4) and (5), we propose a prompt approximation algorithm with the computational complexity of $\mathcal{O}(n \log n)$, in order to use distances of sets to measure the discriminative level of classifiers in practice.

Since the core operation in these two equations is to evaluate the distance between data points inside $\mathcal{X} \times \mathcal{Y}$, to reduce the number of distance evaluation operations involved in them, we observe that the distance between similar data points tends to be closer than others after projecting them onto a general one-dimensional linear subspace. To be concrete, let $g : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ be a random projection, then we could write g as

$$g(\boldsymbol{x}, \ddot{y}; \boldsymbol{w}) = g(\breve{\boldsymbol{x}}, \boldsymbol{a}, \ddot{y}; \boldsymbol{w}) = [\ddot{y}, x_1, ..., x_{n_x}]^\mathsf{T} \boldsymbol{w}\,, \tag{9}$$

where $\boldsymbol{w} = [w_0, w_1, ..., w_{n_x}]^\mathsf{T}$ is a non-zero random vector. Now, we choose a random projection $g : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, and then sort all the projected data points on $\mathbb{R}$. According to (9), it is likely that for the instance $(\boldsymbol{x}, y)$ in S_j, the desired instance $\operatorname{argmin}_{(\boldsymbol{x}', y') \in \bar{S}_j} \mathbf{d}\big((\breve{\boldsymbol{x}}, y), (\breve{\boldsymbol{x}}', y')\big)$ would be somewhere near it after the projection, and vice versa. Thus, by using the projections, we could accelerate the process in (4) and (5) by checking several adjacent instances rather than traversing the whole dataset.

Then we could propose an approximation algorithm to estimate the distance between sets in (4) and (5), named as '*Approximation of distance between sets for one sensitive attribute (ApproxDist)*', shown in Algorithm 2. As for the distance in (6) and (7), we propose '*Approximation of extended distance between sets for several sensitive attributes (ExtendDist)*', shown in Algorithm 3. Note that there exists a sub-route within *ApproxDist* to obtain approximated distances between sets, which is named as '*Acceleration sub-procedure (AcceleDist)*' and shown in Algorithm 1. As the time complexity of sorting in line 2 of Algorithm 1 could reach $\mathcal{O}(n \log n)$, we could get the computational complexity of Algorithm 1 as follows: i) The complexity of line 1 is $\mathcal{O}(n)$; and ii) The complexity from line 4 to line 10 is $\mathcal{O}(2m_2 + 1)$. Thus the overall time complexity of Algorithm 1 would be $\mathcal{O}(n(\log n + m_2 + 1))$, and that of Algorithm 2 be $\mathcal{O}(m_1 n(\log n + m_2))$, and that of Algorithm 3 be $\mathcal{O}(n_a m_1 n(\log n + m_2))$. As both m_1 and m_2 are the designated constants, and n_a is also a fixed constant for one specific dataset,

Algorithm 2 Approximation of distance between sets for one sensitive attribute with binary/multiple values, aka. *ApproxDist* $(\{(\breve{\boldsymbol{x}}_i, a_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n; m_1, m_2)$

Input: Dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, prediction of S by the classifier $f(\cdot)$ that has been trained, that is, $\{\hat{y}_i\}_{i=1}^n$, and two hyper-parameters m_1 and m_2 as the designated numbers for repetition and comparison respectively
Output: Approximation of $\mathbf{D}_{\cdot,\boldsymbol{a}}(S, a_i)$ and $\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S, a_i)$
1: **for** j from 1 to m_1 **do**
2: Take two orthogonal vectors $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ where each $\boldsymbol{w}_k \in [-1, +1]^{1+n_x}$ $(k = \{0, 1\})$
3: **for** k from 0 to 1 **do**
4: $t^k_{\text{max}}, t^k_{\text{avg}} =$ *AcceleDist* $(\{(\breve{\boldsymbol{x}}_i, a_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n, \boldsymbol{w}_k; m_2)$
5: **end for**
6: $d^j_{\text{max}} = \min\{t^k_{\text{max}} \mid k \in \{0, 1\}\} = \min\{t^0_{\text{max}}, t^1_{\text{max}}\}$
7: $d^j_{\text{avg}} = \min\{t^k_{\text{avg}} \mid k \in \{0, 1\}\} = \min\{t^0_{\text{avg}}, t^1_{\text{avg}}\}$
8: **end for**
9: **return** $\min\{d^j_{\text{max}} \mid j \in [m_1]\}$ and $\frac{1}{n}\min\{d^j_{\text{avg}} \mid j \in [m_1]\}$

Algorithm 3 Approximation of extended distance between sets for several sensitive attributes with binary/multiple values, aka. *ExtendDist* $(\{(\breve{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n; m_1, m_2)$,

Input: Dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n = \{(\breve{\boldsymbol{x}}_i, \boldsymbol{a}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{a}_i = [a_{i,1}, a_{i,2}, ..., a_{i,n_a}]^\intercal$, prediction of S by the classifier $f(\cdot)$ that has been trained, that is, $\{\hat{y}_i\}_{i=1}^n$, and two hyper-parameters m_1 and m_2 as the designated numbers for repetition and comparison respectively
Output: Approximation of $\mathbf{D}_{\cdot,\boldsymbol{a}}(S)$ and $\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S)$
1: **for** j from 1 to n_a **do**
2: $d^{(j)}_{\text{max}}, d^{(j)}_{\text{avg}} =$ *ApproxDist* $(\{(\breve{\boldsymbol{x}}_i, a_{i,j})\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n; m_1, m_2)$
3: **end for**
4: **return** $\max_{1 \leqslant j \leqslant n_a}\{d^{(j)}_{\text{max}} \mid j \in [n_a]\}$ and $\frac{1}{n_a}\sum_{j=1}^{n_a} d^{(j)}_{\text{avg}}$

the time complexity of computing the distance is then down to $\mathcal{O}(n \log n)$, which is more welcome than $\mathcal{O}(n^2)$ for the direct computation. Note that a simplified and faster version of *ApproxDist* is also provided in Appendix B.1, with the same computational complexity of $\mathcal{O}(m_1 n(\log n + m_2))$.

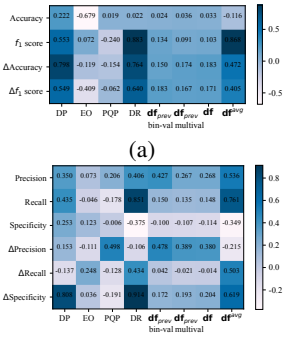
It is worth noting that in line 9 of Algorithm 2, we use the minimal instead of their average value. The reason is that in each projection, the exact distance for one instance would not be larger than the calculated distance for it via *AcceleDist*; and the same observation holds for all of the projections in *ApproxDist*. Thus, the calculated distance via *ApproxDist* is always no less than the exact distance, and the minimal operator should be taken finally after multiple projections. Also note that we demonstrate the algorithmic effectiveness of Algorithm 2 under certain reasonable assumptions in Appendix B.2.

3 Empirical results

In this section, we elaborate on our experiments to evaluate the effectiveness of the proposed *HFM* in (8) as well as *ExtendDist* and *ApproxDist*. More details are elaborated in Appendix C to save space.

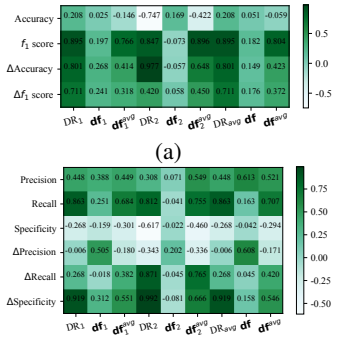
Comparison between *HFM* and baseline fairness measures The aim of this experiment is to evaluate the effectiveness of the proposed *HFM* compared with baseline fairness measures. We compare the correlation (referring to the Pearson correlation coefficient) between the performance difference and different fairness measures, and report the empirical results in Fig. 1–2 and Appendix C.2.

For one single sensitive attribute in Fig. 1, $\mathbf{df}^{\text{avg}}$ is highly correlated with recall/sensitivity and f_1 score. Besides, even $\mathbf{df}^{\text{avg}}$ only describes the extra bias, its correlation with $\Delta(\text{performance})$ is still close to that of DR (and sometimes DP), which means the average *HFM* (i.e., $\mathbf{df}^{\text{avg}}$) can capture the bias within classifiers indeed and that it captures the bias more finely than $\mathbf{df}_{\text{prev}}$ and $\mathbf{df}$. Moreover, $\mathbf{df}^{\text{avg}}$ shows higher correlation with $\Delta(\text{performance})$ than $\mathbf{df}$ in most cases, which means $\mathbf{df}^{\text{avg}}$ may capture the extra bias level of classifiers better than $\mathbf{df}$ in practice. Similar observations could also be found in Fig. 2 for multiple sensitive attributes. Besides, we observe that the correlation between $\mathbf{df}^{\text{avg}}$ and ΔAccuracy (resp. Δf_1 score, ΔSpecificity) achieves half of that of DR, and $\mathbf{df}^{\text{avg}}$ even outperforms DR concerning ΔRecall. Given that *HFM* only captures the extra bias introduced by classifiers, we believe at least $\mathbf{df}^{\text{avg}}$ could capture quite a part of bias within.



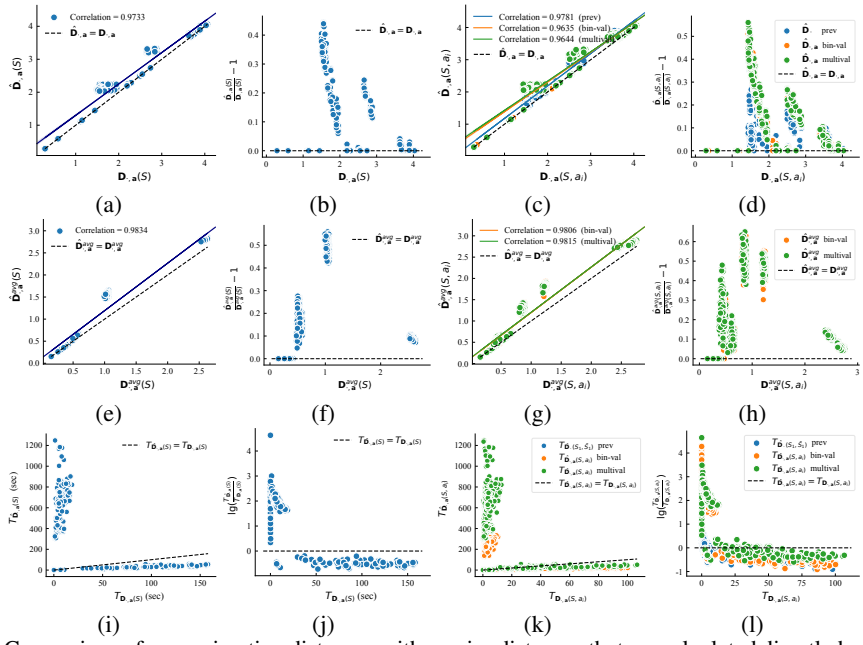
(a) (b)

Figure 1: Correlation heatmap between normal evaluation metric and fairness, for one sensitive attribute. The used notations refer to those in Table 2.



(a) (b)

Figure 2: Correlation heatmap between normal evaluation metric and fairness measure, for all sensitive attributes within the dataset. The notations used here refer to those in Table 3.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 3: Comparison of approximation distances with precise distances that are calculated directly by definition, evaluated on test data. (a–b), (c–d), (e–f), and (g–h) Scatter plots for comparison between approximated and precise values of $\mathbf{D}_{\cdot,\boldsymbol{a}}(S)$, $\mathbf{D}_{\cdot,\boldsymbol{a}}(S,a_i)$, $\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S)$, and $\mathbf{D}^{\text{avg}}_{\cdot,\boldsymbol{a}}(S,a_i)$, respectively; (i–j) Time cost comparison between *ExtendDist* and direct computation; (k–l) Time cost comparison between *ApproxDist* and direct computation. Note that 'prev' denotes approximation results obtained by the simplified Algorithm 4.

Validity of approximation algorithms for distances between sets in Euclidean spaces To verify whether *ApproxDist* and *ExtendDist* could achieve the true distance between sets precisely and timely, we employ scatter plots to compare their values and time cost, presented in Fig. 3 and Appendix C.3.

As shown in Fig. 3(c)–3(d), the approximated values of maximal distance $\mathbf{D}_{\mathbf{a}}(S,a_i)$ are highly correlated with their corresponding precise values. Besides, their linear fit line and the identity line (that is, $f(x)=x$) are near and almost parallel, which means the approximated values are pretty close to their precise value. Similar observations are concluded for the average distance $\mathbf{D}^{\text{avg}}_{\mathbf{a}}(S,a_i)$ in Fig. 3(g)–3(h), maximal distance $\mathbf{D}_{\mathbf{a}}(S)$ in Fig. 3(a)–3(b), and average distance $\mathbf{D}^{\text{avg}}_{\mathbf{a}}(S)$ in Fig. 3(e)–3(f), respectively. As for the execution time of approximation and direct computation in Fig. 3(k)–3(l), *ApproxDist* may take a bit longer time in scenarios of multi-value cases than that of binary values, while all of them could achieve a shorter time than precise values when the execution of direct computation is costly. As for the execution time of approximation and direct computation in Fig. 3(i)–3(j), *ExtendDist* would obtain a bigger advantage when computing precise values is expensive, while on the opposite, we do not need *ExtendDist* that much and can directly calculate them instead.

4 Conclusion

In this paper, we investigate how to evaluate the discrimination level of classifiers in the face of multi-attribute protection scenarios and present a novel harmonic fairness measure with three optional versions, of which all are based on distances between sets from a manifold perspective. To accelerate the computation of distances between sets and reduce its time cost from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$, we further propose two approximation algorithms to resolve bias evaluation in scenarios for single attribute protection and multi-attribute protection, respectively. The empirical results have demonstrated that the proposed fairness measure and approximation algorithms are valid and effective.

References

[1] BAROCAS, Solon ; HARDT, Moritz ; NARAYANAN, Arvind: *Fairness and machine learning: Limitations and opportunities*. MIT Press, 2023

[2] BERK, Richard ; HEIDARI, Hoda ; JABBARI, Shahin ; KEARNS, Michael ; ROTH, Aaron: Fairness in criminal justice risk assessments: The state of the art. In: *Sociol Methods Res* 50 (2021), Nr. 1, S. 3–44

[3] BIAN, Yijun ; ZHANG, Kun ; QIU, Anqi ; CHEN, Nanguang: Increasing Fairness via Combination with Learning Guarantees. In: *arXiv preprint arXiv:2301.10813* (2023)

[4] CHOULDECHOVA, Alexandra: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. In: *Big Data* 5 (2017), Nr. 2, S. 153–163

[5] CHUANG, Ching-Yao ; MROUEH, Youssef: Fair mixup: Fairness via interpolation. In: *ICLR*, 2021

[6] CRUZ, André F ; BELÉM, Catarina ; BRAVO, João ; SALEIRO, Pedro ; BIZARRO, Pedro: FairGBM: Gradient Boosting with Fairness Constraints. In: *ICLR*, 2023

[7] DU, Mengnan ; MUKHERJEE, Subhabrata ; WANG, Guanchu ; TANG, Ruixiang ; AWADALLAH, Ahmed ; HU, Xia: Fairness via representation neutralization. In: *NeurIPS* Bd. 34, 2021, S. 12091–12103

[8] DWORK, Cynthia ; HARDT, Moritz ; PITASSI, Toniann ; REINGOLD, Omer ; ZEMEL, Richard: Fairness through awareness. In: *ITCS*, ACM, 2012, S. 214–226

[9] FELDMAN, Michael ; FRIEDLER, Sorelle A. ; MOELLER, John ; SCHEIDEGGER, Carlos ; VENKATASUBRAMANIAN, Suresh: Certifying and removing disparate impact. In: *SIGKDD*, 2015, S. 259–268

[10] GAJANE, Pratik ; PECHENIZKIY, Mykola: On formalizing fairness in prediction with machine learning. In: *FAT/ML*, 2018

[11] GONG, Sixue ; LIU, Xiaoming ; JAIN, Anil K.: Mitigating face recognition bias via group adaptive classifier. In: *CVPR*, 2021, S. 3414–3424

[12] GUO, Dandan ; WANG, Chaojie ; WANG, Baoxiang ; ZHA, Hongyuan: Learning fair representations via distance correlation minimization. In: *IEEE Trans Neural Netw Learn Syst* (2022)

[13] HARDT, Moritz ; PRICE, Eric ; SREBRO, Nathan: Equality of opportunity in supervised learning. In: *NIPS* Bd. 29, Curran Associates Inc., 2016, S. 3323–3331

[14] HWANG, Sunhee ; BYUN, Hyeran: Unsupervised image-to-image translation via fair representation of gender bias. In: *ICASSP* IEEE (Veranst.), 2020, S. 1953–1957

[15] IOSIFIDIS, Vasileios ; NTOUTSI, Eirini: AdaFair: Cumulative fairness adaptive boosting. In: *CIKM*. New York, NY, USA : ACM, 2019, S. 781–790

[16] JOO, Jungseock ; KÄRKKÄINEN, Kimmo: Gender slopes: Counterfactual fairness for computer vision models by attribute manipulation. In: *FATE/MM*, 2020, S. 1–5

[17] JOSEPH, Matthew ; KEARNS, Michael ; MORGENSTERN, Jamie H. ; ROTH, Aaron: Fairness in learning: Classic and contextual bandits. In: *NIPS* Bd. 29, Curran Associates, Inc., 2016

[18] JUNG, Sangwon ; LEE, Donggyu ; PARK, Taeeon ; MOON, Taesup: Fair feature distillation for visual recognition. In: *CVPR*, 2021, S. 12115–12124

[19] KANG, Jian ; XIE, Tiankai ; WU, Xintao ; MACIEJEWSKI, Ross ; TONG, Hanghang: InfoFair: Information-theoretic intersectional fairness. In: *Big Data* IEEE (Veranst.), 2022, S. 1455–1464

[20] KARKKAINEN, Kimmo ; JOO, Jungseock: FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In: *CVPR*, 2021, S. 1548–1558

[21] KE, Guolin ; MENG, Qi ; FINLEY, Thomas ; WANG, Taifeng ; CHEN, Wei ; MA, Weidong ; YE, Qiwei ; LIU, Tie-Yan: LightGBM: A highly efficient gradient boosting decision tree. In: *NIPS* Bd. 30, 2017, S. 3146–3154

[22] KHALILI, Mohammad M. ; ZHANG, Xueru ; ABROSHAN, Mahed: Fair sequential selection using supervised learning models. In: *NeurIPS* Bd. 34, 2021, S. 28144–28155

[23] LOCATELLO, Francesco ; ABBATI, Gabriele ; RAINFORTH, Thomas ; BAUER, Stefan ; SCHÖLKOPF, Bernhard ; BACHEM, Olivier: On the fairness of disentangled representations. In: *NeurIPS* Bd. 32, 2019

[24] MO, Sangwoo ; KANG, Hyunwoo ; SOHN, Kihyuk ; LI, Chun-Liang ; SHIN, Jinwoo: Object-aware contrastive learning for debiased scene representation. In: *NeurIPS* Bd. 34, 2021, S. 12251–12264

[25] PADALA, Manisha ; GUJAR, Sujit: FNNC: achieving fairness through neural networks. In: *IJCAI*, 2021. – ISBN 9780999241165

[26] PESSACH, Dana ; SHMUELI, Erez: A review on fairness in machine learning. In: *ACM Comput Surv* 55 (2022), Nr. 3, S. 1–44

[27] PLEISS, Geoff ; RAGHAVAN, Manish ; WU, Felix ; KLEINBERG, Jon ; WEINBERGER, Kilian Q.: On fairness and calibration. In: *NIPS* Bd. 30, 2017

[28] RAMASWAMY, Vikram V. ; KIM, Sunnie S. ; RUSSAKOVSKY, Olga: Fair attribute classification through latent space de-biasing. In: *CVPR*, 2021, S. 9301–9310

[29] ROH, Yuji ; LEE, Kangwook ; WHANG, Steven E. ; SUH, Changho: FairBatch: Batch selection for model fairness. In: *ICLR*, 2021

[30] SARHAN, Mhd H. ; NAVAB, Nassir ; ESLAMI, Abouzar ; ALBARQOUNI, Shadi: Fairness by learning orthogonal disentangled representations. In: *ECCV* Springer (Veranst.), 2020, S. 746–761

[31] TIAN, Huan ; LIU, Bo ; ZHU, Tianqing ; ZHOU, Wanlei ; PHILIP, S Y.: MultiFair: Model Fairness With Multiple Sensitive Attributes. In: *IEEE Trans Neural Netw Learn Syst* (2024)

[32] VERMA, Sahil ; RUBIN, Julia: Fairness definitions explained. In: *FairWare*, 2018, S. 1–7

[33] VERMA, Vikas ; LAMB, Alex ; BECKHAM, Christopher ; NAJAFI, Amir ; MITLIAGKAS, Ioannis ; LOPEZ-PAZ, David ; BENGIO, Yoshua: Manifold mixup: Better representations by interpolating hidden states. In: *ICML* PMLR (Veranst.), 2019, S. 6438–6447

[34] WANG, Mei ; DENG, Weihong: Mitigating bias in face recognition using skewness-aware reinforcement learning. In: *CVPR*, 2020, S. 9322–9331

[35] XU, Han ; LIU, Xiaorui ; LI, Yaxin ; JAIN, Anil ; TANG, Jiliang: To be Robust or to be Fair: Towards Fairness in Adversarial Training. In: *ICML* Bd. 139, PMLR, 2021, S. 11492–11501

[36] ZEMEL, Rich ; WU, Yu ; SWERSKY, Kevin ; PITASSI, Toni ; DWORK, Cynthia: Learning fair representations. In: *ICML* PMLR (Veranst.), 2013, S. 325–333

[37] ZHANG, Hongyi ; CISSE, Moustapha ; DAUPHIN, Yann N. ; LOPEZ-PAZ, David: mixup: Beyond Empirical Risk Minimization. In: *ICLR*, 2018

[38] ZHANG, Tao ; ZHU, Tianqing ; GAO, Kun ; ZHOU, Wanlei ; PHILIP, S Y.: Balancing learning model privacy, fairness, and accuracy with early stopping criteria. In: *IEEE Trans Neural Netw Learn Syst* 34 (2021), Nr. 9, S. 5557–5569

[39] ZHAO, Bowen ; XIAO, Xi ; GAN, Guojun ; ZHANG, Bin ; XIA, Shu-Tao: Maintaining discrimination and fairness in class incremental learning. In: *CVPR*, 2020, S. 13208–13217

[40] ŽLIOBAITĖ, Indrė: Measuring discrimination in algorithmic decision making. In: *Data Min Knowl Discov* 31 (2017), Nr. 4, S. 1060–1089

A Related work

In this section, we firstly introduce existing techniques to enhance fairness and then summarise available metrics to measure fairness for ML models in turn.

A.1 Techniques to enhance fairness

Existing mechanisms to mitigate biases and enhance fairness in ML models could be typically divided into three types: pre-processing, in-processing, and post-processing mechanisms, based on when manipulations are applied during model training pipelines. Particularly, recent work on in-processing fairness for DL models mainly falls under two types of approaches: constraint-based and adversarial learning methods [31]. Constraint-based methods usually incorporate fairness metrics directly into the model optimisation objectives as constraints or regularisation terms. For instance, Zemel *et al.* [36], the pioneer in this direction, put demographic parity constraints on model predictions.

Subsequent work also includes using approximations [35] or modified training schemes [25] to improve scalability. Adversarial methods intend to learn representations as fairly as possible by removing sensitive attribute information. In such procedures, additional prediction heads may be introduced for attribute subgroup predictions and the information concerning sensitive attributes would be removed through inverse gradient updating [34, 20] or disentangling features [18, 23, 30, 12]. Other fairness enhancing techniques include data augmentations [24], sampling [29, 22], data noising [38], dataset balancing with generative methods [14, 16, 28], and reweighting mechanisms [39, 11]. Recently, mixup operations [33, 37, 31] are adopted to enhance fairness by blending inputs across subgroups [5, 7]. However, most of these studies focus on protecting one single sensitive attribute and are hardly able to deal with several sensitive attributes all at once. And multi-attribute fairness protection remains relatively rarely explored.

A.2 Existing fairness metrics and multi-attribute fairness protection

The well-known fairness metrics are generally divided into group fairness—such as demographic parity (DP), equality of opportunity (EO), and predictive quality parity (PQP)—and individual fairness [8, 2, 40, 17, 27]. The former mainly focuses on statistical/demographic equality among groups defined by sensitive attributes, while the latter cares more about the principle that 'similar individuals should be evaluated or treated similarly.' However, satisfying fairness metrics all at once is hard to achieve because they are usually not compatible with each other [1]. In practice, it may need to deliberate on the choice of the specified distance in individual fairness [17, 8]. Moreover, the three commonly used group fairness measures (that is, DP, EO, and PQP) can only deal with one single sensitive attribute with binary values. Although extending them to scenarios of one sensitive attribute with multiple values is possible, they are still limited when facing several sensitive attributes at the same time. Recent work includes a newly proposed fairness measure named discriminative risk (DR) [3] that is capable of capturing bias from both individual and group fairness aspects and two fairness frameworks (that is, InfoFair [19] and MultiFair [31]) to deliver fair predictions in face of multiple sensitive attributes. Yet these two fairness frameworks are not measures that could directly evaluate the discrimination level of ML models.

B Supplemental methodology

B.1 Additional approximation algorithm

There is a simplified and faster version of *ApproxDist*, described in Algorithm 4. This major difference is that: while Algorithm 4 takes only one random vector each time, *ApproxDist* of Algorithm 2 take a few orthogonal random vectors each time and do the projection-relevant process for all these orthogonal vectors. The number of these orthogonal vectors could be $n_x + 1$, or smaller (such as two or three) if the practitioners would like to save more time in practice. For instance, we set two orthogonal random vectors in Algorithm 2 at present. Then we take the minimum among all estimated distances. This modification may slightly increase the time cost of approximation a bit compared with the simplified Algorithm 4, yet will still significantly accelerate the execution speed and the effectiveness of the projection algorithm, compared with the direct calculation of distances.

Algorithm 4 Simplified approximation of distance between sets for one sensitive attribute, aka. $\texttt{ApproxDist}\,(\{(\breve{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n; m_1, m_2)$

Input: Dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, prediction of S by the classifier $f(\cdot)$ that has been trained, that is, $\{\hat{y}_i\}_{i=1}^n$, and two hyper-parameters m_1 and m_2 as the designated numbers for repetition and comparison respectively
Output: Approximation of distance $\mathbf{D}.(S_1, \bar{S}_1)$ in (3)

1: **for** j from 1 to m_1 **do**
2: Take a random vector $\boldsymbol{w}$ from the space $\mathcal{W} = \{\boldsymbol{w} = [w_0, w_1, ..., w_{n_x}]^\mathsf{T} \,|\, \sum_{i=0}^{n_x} |w_i| = 1\} \subseteq [-1, 1]^{1+n_x}$
3: $d^j_{\text{max}}, \_ = \texttt{AcceleDist}\,(\{(\breve{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n, \boldsymbol{w}; m_2)$
4: **end for**
5: **return** $\min\{d^j_{\text{max}} \mid j \in [m_1]\}$

B.2 Algorithmic effectiveness analysis of *ApproxDist*

As *ApproxDist* in Algorithm 2 is the core component devised to facilitate the approximation of direct calculation of the distance between sets, in this section, we detail more about its algorithmic effectiveness under some conditions.

We first introduce an important lemma, stated in Lemma 1, that confirms the observation that 'the distance between similar data points tends to be closer than others after projecting them onto a general one-dimensional linear subspace'. It demonstrates by (10) that the probability $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ also goes to zero when the ratio r_1/r_2 goes to zero. Additionally, it is easy to observe that $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ reaches the same order of magnitude as r_1/r_2, and especially, when r_1 equals r_2, $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ could be roughly viewed as $1/2$ for coarse approximation. It means that the breaking probability of the aforementioned statement—similar data points leading to closer distances—tends to increase as r_1 gradually gets closer to r_2. And the profound meaning behind Lemma 1 is that the bigger the gap of lengths between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is, the more effective and efficient our proposed approximation algorithms would be.

Lemma 1. *Let $\boldsymbol{v}_1$ (resp. $\boldsymbol{v}_2$) be a vector in the n-dimensional Euclidean space $\mathbb{R}^n$ with length r_1 (resp. r_2) such that $r_1 \leqslant r_2$. Let $\boldsymbol{w} \subset \mathbb{R}^n$ be a unit vector. We define $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ as the probability that $|\langle \boldsymbol{w}, \boldsymbol{v}_1 \rangle| \geqslant |\langle \boldsymbol{w}, \boldsymbol{v}_2 \rangle|$. Then,*

$$\frac{\sin\phi}{\pi} \cdot \frac{r_1}{r_2} \leqslant \mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2) \leqslant \left(1 + \frac{r_1^2}{r_2^2}\right)^{-1/2} \cdot \frac{r_1}{r_2}\,, \tag{10}$$

where ϕ represents the angle between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

Proof. Notice that $|\langle \boldsymbol{w}, \boldsymbol{v}_1 \rangle| \geqslant |\langle \boldsymbol{w}, \boldsymbol{v}_2 \rangle|$ is equivalent to

$$\langle \boldsymbol{v}_2 - \boldsymbol{v}_1, \boldsymbol{w} \rangle \langle \boldsymbol{v}_1 + \boldsymbol{v}_2, \boldsymbol{w} \rangle \leqslant 0\,. \tag{11}$$

If $\boldsymbol{w}$ satisfies (11), then it lies between two hyperplanes that are perpendicular to $\boldsymbol{v}_1 - \boldsymbol{v}_2$ and $\boldsymbol{v}_1 + \boldsymbol{v}_2$ respectively. Denote by θ the angle between these two hyperplanes (which is equal to the acute angle between $\boldsymbol{v}_2 - \boldsymbol{v}_1$ and $\boldsymbol{v}_1 + \boldsymbol{v}_2$), then $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \theta/\pi$. Moreover,

$$\sin^2\theta = 1 - \cos^2\theta = \frac{4\|\boldsymbol{v}_1\|^2\|\boldsymbol{v}_2\|^2 - 4\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle^2}{(\|\boldsymbol{v}_1\|^2 + \|\boldsymbol{v}_2\|^2)^2 - 4\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle^2}\,. \tag{12}$$

Here $\|\boldsymbol{v}_i\|^2 = \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = r_i^2$ $(i = 1, 2)$ is the square length of the vector. Recall that $\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle = \|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\|\cos\phi$. By (12), we have

$$\frac{\|\boldsymbol{v}_1\|^2}{\|\boldsymbol{v}_2\|^2}\sin^2\phi \leqslant \sin^2\theta \leqslant \frac{4\|\boldsymbol{v}_1\|^2\|\boldsymbol{v}_2\|^2}{(\|\boldsymbol{v}_1\|^2 + \|\boldsymbol{v}_2\|^2)^2}\,. \tag{13}$$

Combining (13) with the fact that $\frac{2}{\pi}\theta \leqslant \sin\theta \leqslant \theta$, we conclude that the probability $\mathbb{P}(\boldsymbol{v}_2, \boldsymbol{v}_2) = \frac{\theta}{\pi}$ satisfies the desired inequalities. □

Our main result in this section is Proposition 2, whereby (15), the efficiency of *ApproxDist* decreases as the scaled density μ of the original dataset increases. Meanwhile, when dealing with large-scale datasets, the more insensitive attributes we have, the more efficient *ApproxDist* is. In general, the efficiency of *ApproxDist* depends on the shape of these two subsets of S. Roughly speaking, the more separated these two sets are from each other, the more efficient *ApproxDist* is.

Proposition 2. *Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be a $(k+1)$-dimensional dataset where instances have $(k+1)$ features, an evenly distributed dataset with a size of n that is a random draw of the feature-label space $\mathcal{X} \times \mathcal{Y}$. For any two subsets of S with distance d (ref. (3)), suppose further that the scaled density*

$$\limsup_{\mathbf{B} \subset \mathbb{R}^{k+1} \text{ an Euclidean ball}} \frac{1}{\mathrm{Vol}(\mathbf{B})}\#(\mathbf{B} \cap S) = \frac{\mu}{\mathrm{Vol}(\mathbf{B}(d))}\,, \tag{14}$$

for some positive real number μ (here $\#$ denotes the number of points of a finite set and $\mathbf{B}(d)$ denotes a ball of radius d). Then, with probability at least

$$1 - \left(\frac{\pi\mu}{m_2\mathrm{Vol}(\mathbf{B}(1))}\left(\left(1 + \frac{n}{\mu}\right)^{\frac{1}{k+1}} - \alpha\right)\right)^{m_1}\,, \tag{15}$$

ApproxDist *could reach an approximate solution that is at most α times of the distance between these two subsets.*

Proof. Let S_0 and S_1 be two sub-datasets of S. We fix the instance $\boldsymbol{v}_0 \in S_0$ such that $d \triangleq \mathbf{D}(S_0, S_1) = \mathbf{d}(\boldsymbol{v}_0, \boldsymbol{v_1})$ for some $\boldsymbol{v_1} \in S_1$. For simplicity, we may set $\boldsymbol{v}_0$ as the origin. The probability that an instance $\boldsymbol{v} \in S_1$ has a shorter length than $\boldsymbol{v}_1$ after projection to a line (see (9)) is denoted as $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v})$. By assumption, we only need to consider those instances whose length is greater than αd (outside the ball $\mathbf{B}(\alpha d)$ centered at origin). Hence, the desired probability is bounded from below by

$$1 - \left(\frac{1}{m_2} \sum_{\boldsymbol{v} \notin \mathbf{B}(\alpha d)} \mathbb{P}(\boldsymbol{v}_0, \boldsymbol{v}) \right)^{m_1}. \tag{16}$$

However, (16) is based on the extreme assumption that all instances lie on the same two-dimensional plane. In our case, the instances are evenly distributed. Hence, we may adjust the probability by multiplying

$$\frac{\mathrm{Vol}(S^1(\frac{\|\boldsymbol{v}\|}{d}))}{\mathrm{Vol}(S^k(\frac{\|\boldsymbol{v}\|}{d}))} = \frac{\Gamma(\frac{k+1}{2})}{\pi^{\frac{k-1}{2}}} \cdot \left(\frac{d}{\|\boldsymbol{v}\|} \right)^{k-1},$$

where $\Gamma(\cdot)$ denotes the Gamma function and $\mathrm{Vol}(S^i(r))$ denotes the area of the i-th dimensional sphere of radius r. Hence, by Lemma 1, the desired probability is lower bounded by

$$1 - \left(\frac{1}{m_2} \sum_{\boldsymbol{v} \notin \mathbf{B}(\alpha d)} \left(1 + \frac{d^2}{\|\boldsymbol{v}\|^2} \right)^{-\frac{1}{2}} \cdot \frac{\Gamma(\frac{k+1}{2})}{\pi^{\frac{k-1}{2}}} \cdot \left(\frac{d}{\|\boldsymbol{v}\|} \right)^k \right)^{m_1}. \tag{17}$$

Under our assumption, (17) attains the lowest value when the data are evenly distributed inside a hollow ball $\mathbf{B}_0 \setminus \mathbf{B}(d)$ centered at $\boldsymbol{v}_0$. The radius of $\mathbf{B}_0$, denoted as r_0, satisfies

$$n - 1 = \mu \frac{\mathrm{Vol}(\mathbf{B}_0 \setminus \mathbf{B}(d))}{\mathrm{Vol}(\mathbf{B}(d))} = \mu \left(\left(\frac{r_0}{d} \right)^{k+1} - 1 \right). \tag{18}$$

In this situation, we may write the summation part of (17) as an integration. To be more specific, (17) is lower bounded by

$$1 - \left(\frac{1}{m_2} \int_{\alpha d}^{r_0} A(x) \mu \mathrm{Vol}(S^k(x)) dx \right)^{m_1}. \tag{19}$$

where $A(x) = (1 + \frac{d^2}{x^2})^{-\frac{1}{2}} \frac{\Gamma(\frac{k+1}{2})}{\pi^{(k-1)/2}} \cdot (\frac{d}{x})^k$. Moreover, (19) can be simplified as

$$1 - \left(\frac{1}{m_2 \mathrm{Vol}(\mathbf{B}(1))} \int_{\alpha d}^{r_0} \frac{\pi \mu}{d} \cdot \frac{x}{\sqrt{x^2 + d^2}} dx \right)^{m_1}. \tag{20}$$

Combining (18) and (20), we conclude that the desired probability is lower bounded by

$$1 - \left(\frac{\pi \mu}{m_2 \mathrm{Vol}(\mathbf{B}(1))} \left(\left(\left(1 + \frac{n}{\mu} \right)^{\frac{2}{k+1}} + 1 \right)^{\frac{1}{2}} - (\alpha^2 + 1)^{\frac{1}{2}} \right) \right)^{m_1}. \tag{21}$$

And the proposition follows from (21). □

Now we discuss the choice of hyper-parameters (i.e., m_1 and m_2) according to (15). In fact, (15) can be approximately written as $1 - c \cdot n^{\frac{m_1}{k+1}} / m_2^{m_1}$. We can calculate the order of magnitude of $n^{\frac{m_1}{k+1}} / m_2^{m_1}$ by taking the logarithm:

$$-\lambda \triangleq \lg \left(n^{\frac{m_1}{k+1}} / m_2^{m_1} \right) = m_1 \left(\frac{\lg n}{k+1} - \lg m_2 \right). \tag{22}$$

Therefore *ApproxDist* could reach an approximate solution with probability at least $(1 - c \cdot 10^{-\lambda})$. In practice, we choose positive integers m_2 and m_1 such that λ is reasonably large, ensuring that the algorithm will reach an approximate solution with high probability.

C Supplemental empirical results

In this section, we elaborate on our experiments to evaluate the effectiveness of the proposed *HFM* in (8) and *ExtendDist* in Algorithm 3, as well as *ApproxDist* in Algorithm 2. These experiments are conducted to explore the following research questions: **RQ1**. Compared with baseline fairness measures, does the proposed *HFM* capture the discrimination level of one classifier effectively, as well as from both individual and group fairness aspects, and can it capture the discrimination level when facing several sensitive attributes with multiple values at the same time? **RQ2**. Can *ApproxDist* approximate the direct computation of distances in (4) and (5) precisely, and how efficient is *ApproxDist* compared with the direct computation? And by extension, can *ExtendDist* approximate the direct computation of distances in (6) and (7) precisely, and how efficient is *ExtendDist*?

C.1 Experimental setups

In this subsection, we present the experimental settings we use, including datasets, evaluation metrics, baseline fairness measures, and implementation details.

Datasets Five public datasets were adopted in the experiments: Ricci,[2] Credit,[3] Income,[4] PPR, and PPVR.[5] Each of them has two sensitive attributes except Ricci, with more details provided in Table 1.

Table 1: Dataset statistics. The column '#inst' represents the number of instances; The $\text{joint}_{\text{both}}$ column represents that both two of the sensitive attribute values of one instance belong to the corresponding privileged group; The $\text{joint}_{\text{either}}$ column represents for this instance, at least one of its sensitive values belongs to the corresponding privileged group.

Dataset	**#inst**	**#feature**		**#member in the privileged group**			
		raw	processed	1st priv	2nd priv	$\text{joint}_{\text{both}}$	$\text{joint}_{\text{either}}$
ricci	118	5	6	68 in race	—	—	—
credit	1000	21	58	690 in sex	851 in age	625	916
income	30162	14	98	25933 in race	20380 in sex	18038	28275
ppr	6167	11	401	4994 in sex	2100 in race	1620	5474
ppvr	4010	11	327	3173 in sex	1452 in race	1119	3506

Evaluation metrics As data imbalance usually exists within unfair datasets, we consider several criteria to evaluate the prediction performance from different perspectives, including accuracy, precision, recall (aka. sensitivity), f_1 score, and specificity. For efficiency metrics, we directly compare the time cost of different methods.

Baseline fairness measures To evaluate the validity of *HFM* in capturing the discriminative degree of classifiers, we compare it with three commonly-used group fairness measures (that is, demographic parity (DP) [9, 10], equality of opportunity (EO) [13], and predictive quality parity (PQP) [4, 32])[6] and discriminative risk (DR)[7] [3] that could reflect the bias level of ML models from both individual- and group-fairness aspects.

Implementation details We mainly use bagging, AdaBoost, LightGBM [21], FairGBM [6], and AdaFair [15] as learning algorithms, where FairGBM and AdaFair are two fairness-aware ensemble-based methods. Plus, certain kinds of classifiers are used in Section 3—including decision trees (DT), naive Bayesian (NB) classifiers, k-nearest neighbours (KNN) classifiers, Logistic Regression (LR), support vector machines (SVM), linear SVMs (linSVM), and multilayer perceptrons (MLP)—so that we have a larger learner pools to choose from based on different fairness-relevant rules. Standard 5-fold cross-validation is used in these experiments, in other words, in each iteration, the entire dataset is divided into two parts, with 80% as the training set and 20% as the test set. Also, features of datasets are scaled in preprocessing to lie between 0 and 1. Except for the experiments for RQ3, we set the hyper-parameters $m_1 = 25$ and $m_2 = \lceil 2\lg(n) \rceil$ in other experiments.

---

[2] `https://rdrr.io/cran/Stat2Data/man/Ricci.html`

[3] `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data`

[4] `https://archive.ics.uci.edu/ml/datasets/adult`

[5] `https://github.com/propublica/compas-analysis/`, that is, Propublica-Recidivism and Propublica-Violent-Recidivism datasets

[6] Three commonly used group fairness measures of one classifier $f(\cdot)$ are evaluated as

$$\text{DP}(f) = |\mathbb{P}_{\mathcal{D}}[f(\boldsymbol{x})=1 | \boldsymbol{a}=1] - \mathbb{P}_{\mathcal{D}}[f(\boldsymbol{x})=1 | \boldsymbol{a}=0]|\,, \tag{23a}$$

$$\text{EO}(f) = |\mathbb{P}_{\mathcal{D}}[f(\boldsymbol{x})=1 | \boldsymbol{a}=1,\ y=1] - \mathbb{P}_{\mathcal{D}}[f(\boldsymbol{x})=1 | \boldsymbol{a}=0,\ y=1]|, \tag{23b}$$

$$\text{PQP}(f) = |\mathbb{P}_{\mathcal{D}}[y=1 | \boldsymbol{a}=1,\ f(\boldsymbol{x})=1] - \mathbb{P}_{\mathcal{D}}[y=1 | \boldsymbol{a}=0,\ f(\boldsymbol{x})=1]|, \tag{23c}$$

respectively, where $\boldsymbol{x} = (\breve{\boldsymbol{x}}, \boldsymbol{a})$, y, and $f(\boldsymbol{x})$ are respectively features, the true label, and the prediction of this classifier for one instance. Note that $\boldsymbol{a} = 1$ and 0 respectively mean that the instance $\boldsymbol{x}$ belongs to the privileged group and marginalised groups.

[7] The discriminative risk (DR) of this classifier is evaluated as

$$\text{DR}(f) = \mathbb{E}_{\mathcal{D}}[\mathbb{I}(f(\breve{\boldsymbol{x}}, \boldsymbol{a}) \neq f(\breve{\boldsymbol{x}}, \tilde{\boldsymbol{a}}))]\,, \tag{24}$$

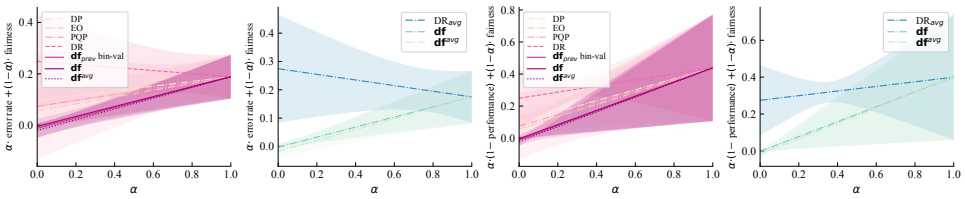
where $\tilde{\boldsymbol{a}}$ represents the disturbed sensitive attributes. DR reflects its bias degree from both individual- and group-fairness aspects.

Table 2: Test evaluation performance of different fairness measures, where LightGBM is used as the learning algorithm. The column named 'Att$_{\text{sen}}$' denotes a corresponding sensitive attribute, and Δ denotes the performance difference between a metric and that after disturbing the data [3]. Note that $\mathbf{df}_{\text{prev}} = \mathbf{D}_f(S_1, \bar{S}_1)/\mathbf{D}(S_1, \bar{S}_1) - 1$ represents the simplified Algorithm 4, and $\mathbf{df} = \log(\mathbf{D}_{f,\boldsymbol{a}}(S, a_i)/\mathbf{D}_{\boldsymbol{a}}(S, a_i))$ and $\mathbf{df}^{\text{avg}} = \log(\mathbf{D}^{\text{avg}}_{f,\boldsymbol{a}}(S, a_i)/\mathbf{D}^{\text{avg}}_{\boldsymbol{a}}(S, a_i))$ here represent *HFM* in this paper for each sensitive attribute.

Dataset	Att$_{\text{sen}}$	Normal evaluation metric				Baseline fairness measure					Proposed fairness measure		
		Accuracy	f_1 score	ΔAccuracy	Δf_1 score	DP	EO	PQP	DR	$\mathbf{df}_{\text{prev}}$ bin-val	$\mathbf{df}_{\text{prev}}$ multival	$\mathbf{df}$	$\mathbf{df}^{\text{avg}}$
ricci	race	99.5789±0.5766	99.5604±0.6019	52.2105±0.5766	35.2747±0.6019	0.3112±0.0424	0.0000±0.0000	0.0121±0.0166	0.5221±0.0058	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0016±0.0022
credit	sex	77.8750±1.1726	86.2892±0.6221	10.2750±3.9906	11.7147±4.7568	0.0189±0.0095	0.0016±0.0006	0.0666±0.0189	0.3438±0.1001	-0.0059±0.0181	-0.0059±0.0181	-0.0026±0.0079	-0.0075±0.0005
	age	77.8750±1.1726	86.2892±0.6221	10.2750±3.9906	11.7147±4.7568	0.0335±0.0137	0.0065±0.0037	0.1107±0.0209	0.3438±0.1001	-0.0047±0.0105	-0.0047±0.0105	-0.0021±0.0046	-0.0073±0.0008
income	race	83.3998±0.2568	51.6536±1.4002	3.8515±3.6332	6.6956±3.6031	0.0395±0.0013	0.0126±0.0050	0.0110±0.0069	0.1542±0.1015	-0.0414±0.0218	-0.0414±0.0218	-0.0185±0.0099	-0.0170±0.0012
	sex	83.3998±0.2568	51.6536±1.4002	3.8515±3.6332	6.6956±3.6031	0.0886±0.0033	0.0793±0.0089	0.0106±0.0063	0.1542±0.1015	-0.0075±0.0160	-0.0075±0.0160	-0.0033±0.0071	-0.0073±0.0007
ppr	sex	70.0507±0.4676	62.9810±1.4929	10.0709±0.3289	1.4437±0.9277	0.1861±0.0207	0.1800±0.0357	0.0169±0.0082	0.3598±0.0100	-0.0040±0.0078	-0.0040±0.0078	-0.0017±0.0034	0.0051±0.0103
	race	70.0507±0.4676	62.9810±1.4929	10.0709±0.3289	1.4437±0.9277	0.1891±0.0272	0.2192±0.0297	0.0377±0.0143	0.3598±0.0100	-0.0134±0.0134	-0.0114±0.0104	-0.0050±0.0046	-0.0154±0.0139
ppvr	sex	83.8953±0.2315	1.9415±2.7688	0.1620±0.2315	1.9415±2.7688	0.0020±0.0029	0.0113±0.0162	0.4000±0.5477	0.0016±0.0023	-0.0107±0.0625	-0.0107±0.0625	-0.0054±0.0268	-0.0560±0.0042
	race	83.8953±0.2315	1.9415±2.7688	0.1620±0.2315	1.9415±2.7688	0.0008±0.0011	0.0048±0.0093	0.0000±0.0000	0.0016±0.0023	-0.0150±0.0930	-0.0027±0.0253	-0.0013±0.0109	-0.0785±0.0036

Table 3: Test evaluation performance of different fairness measures, where LightGBM is used as the learning algorithm. The notation Δ denotes the performance difference between a metric and that after disturbing the data and DR works for one sensitive attribute with multiple values [3]. Here we use $\text{DR}_{avg} = \frac{1}{n_a}\sum_{i=1}^{n_a} \text{DR}_i$ to reflect the bias level on the whole dataset.

Dataset	Normal evaluation metric				Fairness for first sensitive attribute			Fairness for second sensitive attribute			Fairness for all sensitive attributes		
	Accuracy	f_1 score	ΔAccuracy	Δf_1 score	DR_1	$\mathbf{df}_1$	$\mathbf{df}_1^{\text{avg}}$	DR_2	$\mathbf{df}_2$	$\mathbf{df}_2^{\text{avg}}$	DR_{avg}	$\mathbf{df}$	$\mathbf{df}^{\text{avg}}$
ricci	97.3913±2.3814	97.3085±2.4628	49.5652±2.3814	32.6026±2.4628	0.5130±0.0364	0.0000±0.0000	-0.0031±0.0271	—	—	—	0.5130±0.0364	0.0000±0.0000	-0.0031±0.0271
credit	77.8750±1.1726	86.2892±0.6221	10.2750±3.9906	11.7147±4.7568	0.3438±0.1001	-0.0026±0.0079	-0.0075±0.0005	0.3438±0.1001	-0.0021±0.0046	-0.0073±0.0008	0.3438±0.1001	-0.0021±0.0046	-0.0074±0.0005
income	83.3998±0.2568	51.6536±1.4002	3.8515±3.6332	6.6956±3.6031	0.1542±0.1015	-0.0185±0.0099	-0.0170±0.0012	0.1542±0.1015	-0.0033±0.0071	-0.0073±0.0007	0.1542±0.1015	-0.0041±0.0068	-0.0107±0.0005
ppr	70.0507±0.4676	62.9810±1.4929	10.0709±0.3289	1.4437±0.9277	0.3598±0.0100	-0.0017±0.0034	0.0051±0.0103	0.3598±0.0100	-0.0050±0.0046	-0.0154±0.0139	0.3598±0.0100	-0.0017±0.0034	-0.0026±0.0108
ppvr	83.8953±0.2315	1.9415±2.7688	0.1620±0.2315	1.9415±2.7688	0.0016±0.0023	-0.0054±0.0268	-0.0560±0.0042	0.0016±0.0023	-0.0013±0.0109	-0.0785±0.0036	0.0016±0.0023	-0.0054±0.0268	-0.0647±0.0034



(a) (b) (c) (d)

Figure 4: Plots of best test-set fairness-performance trade-offs per fairness metric [6] (the smaller the better). (a) Plot of fairness-accuracy trade-off for one single sensitive attribute; (b) Plot of fairness-accuracy trade-off for all sensitive attributes; (c–d) Plots of fairness-f_1 score trade-off for one sensitive attribute and for all sensitive attributes, respectively. Note that the notations in (a) and (c) refer to those in Table 2, and that in (b) and (d) refer to those in Table 3.

C.2 Comparison between *HFM* and baseline fairness measures

The aim of this experiment is to evaluate the effectiveness of the proposed *HFM* compared with baseline fairness measures. As groundtruth discriminative levels of classifiers remain unknown and it is hard to directly compare different methods from that perspective, we compare the correlation (referring to the Pearson correlation coefficient) between the performance difference and different fairness measures. The empirical results are reported in Figures 1–2 in Section 3, as well as Fig. 4 and Tables 2–3.

For one single sensitive attribute, we can see from Fig. 1 that $\mathbf{df}^{\text{avg}}$ is highly correlated with recall/sensitivity and f_1 score. Besides, even $\mathbf{df}^{\text{avg}}$ only describes the extra bias, its correlation with $\Delta(\text{performance})$ is still close to that of DR (and sometimes DP), which means the average *HFM* (i.e., $\mathbf{df}^{\text{avg}}$) can capture the bias within classifiers indeed and that it captures the bias more finely than $\mathbf{df}_{\text{prev}}$ and $\mathbf{df}$. Moreover, $\mathbf{df}^{\text{avg}}$ shows higher correlation with $\Delta(\text{performance})$ than $\mathbf{df}$ in most cases, which means $\mathbf{df}^{\text{avg}}$ may capture the extra bias level of classifiers better than $\mathbf{df}$ in practice.

As for multiple sensitive attributes, we can see from Fig. 2 that $\mathbf{df}^{\text{avg}}$ is highly correlated with recall/sensitivity and f_1 score and that $\mathbf{df}^{\text{avg}}$ shows higher correlation with $\Delta(\text{performance})$ than $\mathbf{df}$ in most cases, which is similar to our observation in Fig. 1. Note that the original DR [3] calculates all sensitive attributes with binary or multiple values as a whole, and for comparison with *HFM*, we calculate here DR_i for each sensitive attribute and $\text{DR}_{\text{avg}} = \frac{1}{n_a}\sum_{i=1}^{n_a}\text{DR}_i$, analogously to $\mathbf{df}^{\text{avg}}$. Besides, we observe that the correlation between $\mathbf{df}^{\text{avg}}$ and ΔAccuracy (resp. Δf_1 score, ΔSpecificity) achieves half of that of DR, and $\mathbf{df}^{\text{avg}}$ even outperforms DR concerning ΔRecall. Given that *HFM* only captures the extra bias introduced by classifiers, we believe at least $\mathbf{df}^{\text{avg}}$ could capture quite a part of bias within.

Furthermore, we report plots of fairness-performance trade-offs per fairness measure in Fig. 4. We can see that: 1) for one single sensitive attribute, *HFM* (i.e., $\mathbf{df}$ and $\mathbf{df}^{\text{avg}}$) achieves the best result in Fig. 4(a) and 4(c); and 2) for all sensitive attributes on one dataset, $\mathbf{df}$ and $\mathbf{df}^{\text{avg}}$ perform closely and both outperform DR_{avg} in Fig. 4(b) and 4(d). This observation demonstrates the effectiveness of *HFM* from another perspective, in other words, *HFM* could work well if fairness-performance trade-offs need to be considered.

C.3 Validity of approximation algorithms for distances between sets in Euclidean spaces

In this subsection, we evaluate the performance of the proposed *ApproxDist* and *ExtendDist* compared with the precise distance that is directly calculated by definitions. To verify whether they could achieve the true distance between sets precisely and timely, we employ scatter plots to compare their values and time cost, presented in Fig. 3. Note that $\mathbf{D}_{\mathbf{a}}(S, a_i)$ and $\mathbf{D}^{\text{avg}}_{\mathbf{a}}(S, a_i)$ are computed together in *ApproxDist* at one time, and so are $\mathbf{D}_{\mathbf{a}}(S)$ and $\mathbf{D}^{\text{avg}}_{\mathbf{a}}(S, a_i)$ in *ExtendDist*. Also notice that the simplified Algorithm 4 is included for comparison to its current version in scenarios of binary values.

Validity of *ApproxDist* As we can see from Figures 3(c) and 3(d), the approximated values of maximal distance $\mathbf{D}_{\mathbf{a}}(S, a_i)$ are highly correlated with their corresponding precise values. Besides, their linear fit line and the identity line (that is, $f(x) = x$) are near and almost parallel, which means

the approximated values are pretty close to their precise value. Similar observations are concluded for the average distance $\mathbf{D}_{\mathbf{a}}^{\text{avg}}(S, a_i)$ shown in Figures 3(g) and 3(h). As for the execution time of approximation and direct computation in Figures 3(k) and 3(l), *ApproxDist* may take a bit longer time in scenarios of multi-value cases than that of binary values, while all of them could achieve a shorter time than precise values when the execution of direct computation is costly.

Validity of *ExtendDist* As we can see from Figures 3(a) and 3(b), the approximated values of maximal distance $\mathbf{D}_{\mathbf{a}}(S)$ are highly correlated with their corresponding precise values. Besides, their linear fit line and the identity line are near and almost parallel, which means the approximated values are pretty close to their precise value. Similar observations are concluded for the average distance $\mathbf{D}_{\mathbf{a}}^{\text{avg}}(S)$ shown in Figures 3(e) and 3(f). As for the execution time of approximation and direct computation in Figures 3(i) and 3(j), *ExtendDist* would obtain a bigger advantage when computing precise values is expensive, while on the opposite, we do not need *ExtendDist* that much and can directly calculate them instead.