Regression Model Bias Evaluation by Estimating Conditional Densities with Gaussian Mixtures

Wei Sun
AI Center
Verizon Communications
Ashburn, Virginia, USA 20147
wei.sun@verizon.com

Xuning Tang
AI Center
Verizon Communications
Ashburn, Virginia, USA 20147
mike.tang@verizon.com

Kuo-chu Chang
Dept. of SEOR
George Mason University
Fairfax, Virginia, USA 22030
kchang@gmu.edu

***Abstract*—The exploration of AI fairness has emerged as a crucial area of research receiving growing attention in recent years. Various metrics have been proposed to assess group fairness, which examines whether the model outcomes correlate with sensitive attributes such as gender, ethnicity, age, and others. These fairness metrics primarily assess the statistical independence and conditional independence between the model prediction and the true target variable concerning the sensitive attributes. In the fair AI literature, these relationships can generally be categorized into three criteria: Independence, Separation, and Sufficiency, each intuitively defined based on different conditioning variables accordingly. Calculating fairness metrics for classification models is relatively straightforward using confusion matrices. However, it becomes more challenging for regression models due to the continuous nature of the dependent variable and the involvement of probability density function. Previous works on algorithmic fairness often simplify or use less-than-ideal versions of fairness criteria in regression settings. In this paper, we propose a novel approach to calculate Independence, Separation, and Sufficiency scores directly on density level for regression models. We achieve this by estimating the relevant conditional densities with Gaussian mixtures and directly applying them to group fairness approximation. This approach offers greater accuracy when dealing with continuous outputs compared to transformation methods. We validate our approach through empirical studies using both simulated and public datasets. Comparative performance analysis against the most recent existing method demonstrates the effectiveness of our algorithm.**

***Index Terms*—Fair AI, Fairness metric, Regression model, Gaussian mixture, Density estimation, Mutual information.**

I. Introduction

With the extensive use of Artificial Intelligence (AI) and Machine Learning (ML) in various industries to improve business outcomes, it is crucial to ensure proper governance of AI systems. Bias can easily infiltrate these systems and result in irreversible harm. For instance, in loan applications, historically disadvantaged groups may face higher rejection rates compared to privileged groups [1]. More serious cases can be found in court decisions regarding criminal sentences or parole, as exemplified by the well-known case of the COMPAS software reported by ProPublica [2]. Judges in the U.S. often use COMPAS to determine whether to release an offender or keep them in prison, and investigations have revealed bias against African-Americans.

Among the various types of biases, model fairness is a major concern that has garnered increasing attention from academia, industries, and governments in recent years. Detecting model bias has become increasingly important in fields such as healthcare, finance, legal issues, education, and HR. However, there is no universally accepted definition of fairness in the context of machine learning. Two widely adopted families of fairness definitions are group fairness and individual fairness. Group fairness metrics assess fairness at a group level, typically defined by sensitive attributes like age, gender, or ethnicity, while individual fairness metrics aim to ensure that similar individuals receive similar treatment. This paper specifically focuses on group fairness metrics, which are extensively used in existing bias mitigation efforts to quantify model fairness. These metrics are essential for the effective implementation of bias mitigation techniques.

In the fair AI literature, the main criteria of group fairness are commonly known as Independence, Separation, and Sufficiency [3]. These criteria primarily assess the statistical independence and conditional independence relationships between the model prediction, the true target variable, and the sensitive attributes. Under these three fairness principles, various fairness metrics can be defined to quantify the model bias. While group fairness metrics have been well-defined for classification models over the last decade, with biases typically represented by confusion matrices for each demographic group, calculating the same set of group fairness metrics for a regression model poses challenges due to the continuous nature of both the target dependent variable and the model prediction variable.

Prior works on computing group fairness metrics for regression models have often resorted to simplification or less-than-ideal forms of fairness criteria. For example, Agarwal et al. (2019) [4] incorporated statistical parity and bounded group loss as fairness metrics and integrated them into the regression objective function to achieve "fair regression" based on their proposed metrics. Berk et al. (2017) [5] defined a fairness penalty function using extensive pairwise differences in regression predictions between sensitive groups as fairness regularizers. Fitzsimons et al. (2019) [4] suggested incorporating the group fairness expectation into regression kernel functions. Caton and Hass (2020) [5] conducted a review of other fairness metrics focusing on specific parities.

However, none of these studies utilized the rigorous definitions of Independence, Separation, and Sufficiency for calculating fairness in regression models.

To the best of our knowledge, the work by Steinberg, et al. (2020) [6] is the most closely related to our approach. They proposed a method to approximate the three group fairness criteria for regression models by converting the problem into discrete space. Their derivation yielded formulae approximating regression model biases, which could be expressed as ratios of conditional probabilities of the sensitive attributes given the continuous dependent variable, or model prediction or both. They then used machine learning classifiers to estimate these conditional discrete probabilities from the data. Further, these learned probabilities were utilized to calculate mutual information scores, serving as approximations of the regression model bias. Their work was the first to leverage mutual information as a measure to quantify group fairness. However, it is important to note that their approach heavily depends on the performance of the learned classifiers.

Instead of transforming density problem into probability estimation, this paper proposes the use of Gaussian mixture models to directly estimate the conditional densities. This approach enables us to approximate regression fairness at the continuous distribution level. As well known, a Gaussian mixture model can theoretically achieve arbitrary accuracy in estimating any continuous density with an adequate number of components. Please note that in the specific context of group fairness with discrete sensitive attributes like gender or race, we are dealing with only two-dimensional continuous space, namely, the true target variable and the regression model prediction variable. Exploiting this setting is advantageous as we can avoid the curse of dimensionality. To our best knowledge, this paper is the first to apply continuous density estimation to quantify group fairness criteria for regression models. We will present a rigorous mathematical derivation on how we approximate the regression fairness scores, directly computed using estimated densities. Our proposed approach represents an integration of mature algorithms, but it is innovative in the way it assembles them together for the first time for the purpose of fairness evaluation.

The remainder of this paper is organized as follows. In Section II, we provide a formal definition of the problem and introduce the notations that will be used throughout the paper. Section III elaborates our approach, detailing how we utilize Gaussian mixture models to estimate conditional densities and compute Mutual Information as measurements of the corresponding fairness criteria. In Section IV, we present the results of empirical studies conducted on both simulated and public datasets to demonstrate the effectiveness of our approach. Finally, we conclude the paper with a discussion of our findings and outline potential avenues for future work in Section V.

II. Problem Formulation

A. Definitions and Notations

In regression, we have Y and $\hat{Y}$ as the continuous random variables representing the true target and the model prediction respectively. A is a categorical sensitive attribute, e.g. gender, ethnicity, etc., upon which we evaluate model fairness. Typically in regression, $Y, \hat{Y}$, and A are one-dimension random variables. Instances of $Y, \hat{Y}$, and A are denoted as $y, \hat{y}$, and a, which are scalar values in the context. Further, $y \in \mathcal{R}, \hat{y} \in \mathcal{R}$, and $a \in \{1, ..., C\}$ where C represents the cardinality of the categorical variable A. Our data is composed of observations of these three variables. Each data point indexed by i has its corresponding values, $y_i, \hat{y}_i$, and a^i.

Let us review the definition of three most general group fairness criteria: Independence, Separation, and Sufficiency [3]. Specifically, Independence analyzes the statistical independence between model prediction and sensitive attributes. Separation evaluates the conditional independence between model prediction and sensitive attributes given the ground truth. And Sufficiency, on the other hand, evaluates the conditional independence between the true target variable and sensitive attributes given model prediction. Mathematically, these criteria are defined as follows:

Independence :

$$\hat{Y} \perp A \Rightarrow P(\hat{Y}, A) = P(\hat{Y})P(A) \tag{1}$$

Separation :

$$\hat{Y} \perp A \,|Y \Rightarrow P(\hat{Y}, A \,|Y) = P(\hat{Y}|Y)P(A|Y) \tag{2}$$

Sufficiency :

$$Y \perp A \,|\hat{Y} \Rightarrow P(Y, A \,|\hat{Y}) = P(Y|\hat{Y})P(A|\hat{Y}) \tag{3}$$

Knowing that it's very difficult to achieve the perfect equality of these equations in reality, our goal in this paper is to find an effective and robust way to generate a score between zero and one which quantifies to what degree these three fairness criteria are satisfied.

B. Fairness Quantified by Mutual Information

In this section we will first introduce how mutual information can be applied for evaluating fairness based on the group fairness criteria shown in Equations 1 - 3. We then will dive into how to calculate various fairness scores technically.

Mutual Information (MI) or conditional MI of two random variables quantifies how much information is obtained about one variable by observing the other. Therefore it can be used to evaluate independence or conditional independence between two random variables. MI is symmetric by its definition.

Take $\hat{Y}, A$ as the example, MI of $\hat{Y}, A$ can be calculated as below:

$$I[\hat{Y}; A] = \int_{\hat{Y}} \sum_{a \in A} P(\hat{Y}, A) \ln \frac{P(\hat{Y}, A)}{P(\hat{Y})P(A)} d\hat{Y} \tag{4}$$

As you can see, if $\hat{Y}, A$ are independent with each other, their MI $I[Y, A]$ will be zero. Otherwise $I[Y, A]$ will be a positive number.

To make it more meaningful as a metric, it is better to have MI bounded between zero and one. This can be done by using entropy. We know MI is related to entropy in the following way:

$$I[\hat{Y}; A] = H[A] - H[A|\hat{Y}] = H[\hat{Y}] - H[\hat{Y}|A] \tag{5}$$

where $H[A], H[\hat{Y}]$ are entropies of A and $\hat{Y}$ respectively. And $H[A|\hat{Y}], H[\hat{Y}|A]$ are conditional entropies correspondingly. Either $H[A]$ or $H[\hat{Y}]$ can be used as the normaliser then we can make a normalized MI serving as a fairness metric like below:

$$\tilde{I}[\hat{Y}; A] = \frac{I[\hat{Y}; A]}{H[A]} \tag{6}$$

or,

$$\tilde{I}[\hat{Y}; A] = \frac{I[\hat{Y}; A]}{H[\hat{Y}]} \tag{7}$$

Please note that depending on which normaliser is used, the normalized MI may be different. But obviously, $\tilde{I}[Y, A] \in [0, 1]$. We will discuss how to choose the better normaliser later in section III-A.

Similarly, conditional mutual information can be defined and used to evaluate conditional independence. Take $p(\hat{Y}, A|Y)$ as the example, the conditional MI to evaluate the conditional independence between $\hat{Y}, A$ given Y is,

$$I[\hat{Y}; A|Y] = \int_y \int_{\hat{y}} \sum_{a \in A} p(y, \hat{y}, a) \ln \frac{p(\hat{y}, a\,|y)}{p(\hat{y}|y)P(a|y)} d\hat{y}dy \tag{8}$$

And the normaliser for the conditional MI is the corresponding conditional entropy - either $H[A|Y]$ or $H[\hat{Y}|Y]$.

To make this section self sufficient, let us also have the equations for calculating entropy and conditional entropy as below:

$$H[A] = -\sum_{a \in A} P(a) \ln P(a) \tag{9}$$

$$H[Y] = -\int_y p(y) \ln P(y) dy \tag{10}$$

$$H[A|Y] = -\int_y \sum_{a \in A} P(y, a) \ln P(a|y) dy \tag{11}$$

$$H[\hat{Y}|Y] = -\int_{\hat{y}} \int_y P(\hat{y}, y) \ln p(\hat{y}|y) dy d\hat{y} \tag{12}$$

For all these mutual information scores and associated entropies presented above, we will estimate the conditional densities directly by Gaussian mixture models. We will present our algorithm in detail in the next section.

III. Mutual Information by Gaussian Mixtures

A. Minimum upper bound normaliser

To normalize quantities into the interval of $[0, 1]$, it is desirable to find the minimum upper bound among all possible quantities. This is because if an arbitrarily large number is chosen, all original quantities will be compressed towards the lower end of the interval, leaving a large empty space in the upper end. In general, the entropy for discrete variables is well bounded, while the entropy for continuous variables or the conditional entropy for continuous variables may be infinite or may not even exist [10]. In fact, continuous entropy is referred to as differential entropy, and conditional continuous entropy is referred to as conditional differential entropy. Unlike probabilities, which are always in the range of $[0, 1]$, the values of a probability density function can be greater than 1. This means that differential entropy does not share all the properties of discrete entropy. For example, the uniform distribution $\mathcal{U}(0, 1/2)$ has a negative differential entropy: $\int_0^{\frac{1}{2}} -2\ln(2)dx = -\ln(2)$. And the uniform distribution $\mathcal{U}(0, 1)$ has a differential entropy of zero. Furthermore, computing discrete entropy is always easier compared to continuous entropy. Therefore, as a general guideline, it is recommended to choose the entropy of discrete variables as the normalizer for normalizing mutual information scores.

B. Gaussian Mixture Model

Theoretically, a Gaussian mixture model (GMM) can be used to estimate any arbitrary continuous distribution with arbitrary accuracy by using sufficient number of components in the mixture. In practice, the good news is that we often do not need many components to achieve high accuracy as reported in the literature [11], [12].

For calculating mutual information shown in Section II-B, we need to estimate a conditional density of one continuous variable given another continuous variable. This can be done by estimating the joint density over these two variables first. As well known, density estimation suffers by curse of dimensionality. But here in this particular context, the joint continuous space is only up to two dimensions. All group metrics we discussed in the regression setting have only $y, \hat{y}, a$ involved, where a is the discrete sensitive attribute representing sensitive groups. Using Gaussian mixtures can take advantage of this setting to achieve reasonable good performance. Once we have the joint density estimated in the form of a two-dimensional GMM, the conditional density of one dimension given another can be analytically calculated as a one-dimensional GMM based on the original Gaussian components of the joint mixture [13]–[15].

Specifically, suppose that we have the following Gaussian mixture model consisting of K components over the space of $\mathbf{x}_1, \mathbf{x}_2$:

$$p(\mathbf{x}_1, \mathbf{x}_2) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}; \begin{pmatrix} \mu_1^k \\ \mu_2^k \end{pmatrix}, \begin{bmatrix} \Sigma_{11}^k & \Sigma_{12}^k \\ \Sigma_{21}^k & \Sigma_{22}^k \end{bmatrix} \right) \tag{13}$$

where $\mathbf{x}_1 \in \mathcal{R}^{D_1}, \mathbf{x}_2 \in \mathcal{R}^{D_2}$, and k indexes the number of Gaussian components in the mixture. w_k represents the weight of the k^{th} Gaussian component. And Σ^k's are the covariance matrices of $\mathbf{x}_1, \mathbf{x}_2$ for the k^{th} component.

The conditional density of $\mathbf{x}_1$ given $\mathbf{x}_2$ can be derived from Equation 13 with the same number of components Gaussian mixture as below:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \sum_{k=1}^{n} m_k \mathcal{N}(\mathbf{x}_1; \mu_{1|2}^k, \Sigma_{1|2}^k) \tag{14}$$

where m_k is the new mixing coefficient represents the weight for the k^{th} component of the conditional density. And $\mu_{1|2}^k, \Sigma_{1|2}^k$ are the new mean vector and covariance matrix of the k^{th} Gaussian component in the new conditional mixture. Using standard properties of Gaussian distribution (for details, see [15]), they can be calculated by the following equations,

$$\mu_{1|2}^k = \mu_1^k + \Sigma_{12}^k(\Sigma_{22}^k)^{-1}(\mathbf{x}_2 - \mu_2^k) \tag{15}$$

$$\Sigma_{1|2}^k = \Sigma_{11}^k - \Sigma_{12}^k(\Sigma_{22}^k)^{-1}\Sigma_{21}^k \tag{16}$$

$$m_k = \frac{w_k \phi(\mathbf{x}_2|\mu_2^k, \Sigma_{22}^k)}{\sum_{k=1}^{K}(w_k \phi(\mathbf{x}_2|\mu_2^k, \Sigma_{22}^k))} \tag{17}$$

Please note that $\phi(\mathbf{x}_2|\mu_2^k, \Sigma_{22}^k)$ is the likelihood of observed value of $\mathbf{x}_2$ from the k^{th} Gaussian component of the variable $\mathbf{x}_2$.

C. Normalized MI by GMMs for Group Fairness Criteria

As shown in section II-B when calculating the normalized mutual information measures, densities and conditional densities are involved. To be specific, the following densities are needed: (1) $p(\hat{y}), p(\hat{y}, a)$ for Independence; (2) $p(\hat{y}, a|y), p(\hat{y}|y)$ for Separation; and (3) $p(y, a|\hat{y}), p(y|\hat{y})$ for Sufficiency. In addition, discrete probabilities $P(a|y), P(a|\hat{y})$ are also needed for calculating conditional entropies for normalizers. We will show how to estimate all of them by Gaussian mixtures one by one in this section.

*1) **Independence**::* For calculating MI for Independence criteria, we need to estimate $p(\hat{y})$ and $p(\hat{y}|a)$ for each individual value of A. Let us denote a Gaussian mixture as φ. We need to train $\varphi(\hat{y}|a)$ for each unique value of A and $\varphi(\hat{y})$ from all observations. These can be done by one-dimensional Gaussian mixture estimation. Concretely, we have,

$$\begin{aligned} I[\hat{Y}; A] &= \int_{\hat{y}} \sum_{a \in A} p(\hat{y}, A) \ln \frac{p(\hat{y}|a)}{p(\hat{y})} d\hat{y} \\ &\approx \frac{1}{n} \sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|a^i)}{\varphi(\hat{y}_i)} \\ \tilde{I}_{\textbf{Ind}} = \tilde{I}[\hat{Y}; A] &= \frac{I[\hat{Y}; A]}{H[A]} \approx \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|a^i)}{\varphi(\hat{y}_i)}}{-\sum_{a \in A}(\frac{n_a}{n} \ln \frac{n_a}{n})} \end{aligned} \tag{18}$$

*2) **Separation**::* Again, Separation is to evaluate the conditional independence between $\hat{Y}$ and A given Y. First, we have,

$$\begin{aligned} I[\hat{Y}; A|Y] &= \int_y \int_{\hat{y}} \sum_{a \in A} p(y, \hat{y}, a) \ln \frac{p(\hat{y}, a\,|y)}{p(\hat{y}|y)P(a|y)} d\hat{y} dy \\ &= \int_y \int_{\hat{y}} \sum_{a \in A} p(y, \hat{y}, a) \ln \frac{p(\hat{y}|y, a)}{P(\hat{y}|y)} d\hat{y} dy \end{aligned}$$

This requires us to estimate $p(\hat{y}|y, a), p(\hat{y}|y)$ where two continuous variables are engaged. We use the conditional GM estimates $\varphi(\hat{y}|y)$ to approximate $p(\hat{y}|y)$ by the following two steps: (1) train a two-dimensional GM $\varphi(\hat{y}, y)$ from the data with a reasonable number of Gaussian components (usually a $3-5$ components GM works well); (2) Derive the conditional GM $\varphi(\hat{y}|y)$ using Equations 14 - 17 from $\varphi(\hat{y}, y)$. Similarly, $\varphi(\hat{y}|y, a)$ can be trained to approximate $p(\hat{y}|y, a)$ with the corresponding data for each instance of A.

As for the normaliser, we will choose $H[A|Y]$. We know,

$$P(a|y) = \frac{p(y|a)P(a)}{p(y)}$$

Plugging back to Equation 11, we then have,

$$\begin{aligned} H[A|Y] &= -\int_y \sum_{a \in A} p(y, a) \ln P(a|y) dy \\ &\approx -\frac{1}{n} \sum_{i=1}^{n} \ln \frac{\varphi(y_i|a^i)P(a^i)}{\varphi(y_i)} \end{aligned}$$

where $\varphi(y|a)$ and $\varphi(y)$ are one-dimensional GMs that can be easily learned from data, similar to the tasks of $\varphi(\hat{y}|a), \varphi(\hat{y})$ for Independence fairness above.

Now it is ready to present the complete formulae of computing normalized MI for Separation score as below:

$$\begin{aligned} \tilde{I}_{\textbf{Sep}} = \tilde{I}[\hat{Y}; A|Y] &= \frac{I[\hat{Y}; A|Y]}{H[A|Y]} \\ &\approx \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|y_i, a^i)}{\varphi(\hat{y}_i|y_i)}}{-\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(y_i|a^i)p(a^i)}{\varphi(y_i)}} \end{aligned} \tag{19}$$

*3) **Sufficiency**::* Sufficiency criteria is about evaluating the conditional independence between Y and A given $\hat{Y}$. It is very similar to the estimation process for Separation, except the conditional order of variables is the opposite ($p(y|\hat{y})$ instead of $p(\hat{y}|y)$). Accordingly, conditional GM estimates of $\varphi(y|\hat{y}), \varphi(y|\hat{y}, a)$ need to be learned.

Note that the normalizer for Sufficiency is $H[A|\hat{Y}]$, and it is approximated as,

$$\begin{aligned} H[A|\hat{Y}] &= -\int_{\hat{y}} \sum_{a \in A} p(a, \hat{y}) \ln P(a|\hat{y}) d\hat{y} \\ &\approx -\frac{1}{n} \sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|a^i)P(a^i)}{\varphi(\hat{y}_i)} \end{aligned}$$

where $\varphi(\hat{y}), \varphi(\hat{y}|a)$ have been learned in Independence fairness evaluation above.

Now let us present the complete formulae of computing normalized MI for Sufficiency score as below:

$$\tilde{I}_{\textbf{Suf}} = \tilde{I}[Y; A|\hat{Y}] = \frac{I[Y; A|\hat{Y}]}{H[A|\hat{Y}]} \approx \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(y_i|\hat{y}_i, a^i)}{\varphi(y_i|\hat{y}_i)}}{-\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|a^i)p(a^i)}{\varphi(\hat{y}_i)}} \quad (20)$$

*4) **NorMIX Algorithm**::* Now it is ready to present our algorithm - Normalized MIs by Gaussian mixtures as fairness metrics. We named this algorithm as NorMIX. It takes a data set consisting of instances of true target variable, model prediction and sensitive attributes as the input then calculates three fairness scores representing regression fairness in Independence, Separation, and Sufficiency criteria, denoted as $\tilde{I}_{\textbf{Ind}}, \tilde{I}_{\textbf{Sep}}$, and $\tilde{I}_{\textbf{Suf}}$. The complete NorMIX algorithm is depicted in Algorithm 1.

IV. Empirical Studies

A. Simulated data

For a fair comparison, we conducted exactly the same simulation setting demonstrated in [6] and added two more new scenarios. Key variables in the simulation are $a, y, \hat{y}$, where $a \sim Bernoulli(p = 0.7)$, representing a binary sensitive attribute, and $y, \hat{y}$ representing the underlying true target, and the model output, respectively. The data generation process, variables relationship graphs represented by Bayesian networks, and the scatter plots of $y, \hat{y}$ for Cases (a)-(d) are depicted in Fig. 1, which are exactly the same as Cases (a) to (d) in [6] (Section 5.1). Additionally, Cases (e) and (f) were purposely designed to illustrate scenarios with high biases, and zero Separation in the former and zero Sufficiency in the latter. Fig. 2 depicts the distributions and network structures for the two new cases. As can be seen, the conditional independence structures from the network graphs justify the zero fairness scores. In Case (e), for example, $\hat{y}$ is conditionally independent of a given y implying a Separation score of zero. Similarly, in Case (f), y is conditionally independent of a given $\hat{y}$, resulting in a Sufficiency score of zero.

For each case, the true distributions for all variables are known, and therefore the formulae to compute the true fairness scores can be derived accordingly. We approximate the ground truth using numerical integration with varying sample sizes ranging from $10,000$ to 10-million. It was observed that the approximation converges to a stable value for each case. We considered the final converged value as the approximate true fairness scores, which were used as benchmarks for performance comparison. We have documented a detailed technical report including derivation and numerical integration approximation with convergence demonstration.

The simulation in [6] utilized $1,000$ samples for each case and used logistic regression with random radial basis functions to construct non-linear classifiers $\rho(a|\cdot)$, which were validated using 10-fold cross validation. To evaluate the NorMIX algorithm, we adopted the same sample size of $1,000$ and learned the necessary GMMs $\varphi(y), \varphi(\hat{y}), \varphi(y|a), \varphi(\hat{y}|a), \varphi(y, \hat{y})$, and $\varphi(y, \hat{y}|a)$ accordingly. The final fairness scores were computed as the average of the corresponding scores from 10 simulation trials.

For Cases (a)-(d), we presented a side-by-side performance comparison in Table I, with the benchmark provided by the approximate true scores, along with the NorMIX fairness scores, and the corresponding fairness scores in [6] reported originally from their paper. As can be seen, NorMIX produced one small negative normalized MI scores and several scores slightly higher than 1. These are numerical artifacts from the estimation of random processes, which was also mentioned in [6]. We interpret this as fairness scores that are very close to the metric's boundary of 0 or 1. Based on the comparison, it is clear that NorMIX outperforms the other method in terms of accuracy, despite using an identical sample size of $1,000$. The results show that NorMIX was able to reduce the overall Root Mean Square Error (RMSE) by more than half: 0.0293 vs. 0.0755.

Table II presents the performance results for Cases (e) and (f), where the NorMIX scores are compared directly to the approximate ground truth. In these cases, NorMIX was able to detect the conditional independence with very low scores close to zero. Moreover, the overall RMSE of 0.0068 indicates a high level of accuracy for these particular scenarios.

Overall, the NorMIX outcomes align with the analytically intuitive expectation of the simulation design. For example, near-zero score values for Case (a) indicate the fact that the case is of a fair situation. And the high Separation and Sufficiency scores for Case (b) indicate a significant bias issue compared to Case (d), despite both cases having the same network structure but differing in their degree of bias. This is due to the clear disparity observed in Case (b) in terms of the distinguished distribution means, resulting in the separation of the $y, \hat{y}$ scatters into two distinct clusters. In contrast, Case (d) exhibits two groups mixed together with just different distribution variances. Please refer to Fig. 1 (b) and (d) to observe these effects graphically. Interestingly, for Case (c), the almost zero Independence score indicates that $\hat{y}, a$ are independent of each other when not involving y. However, it should be noted that they become conditionally dependent with each other given y. The network structure also provides analytical evidence of this. Similarly, the extremely high degree of conditional dependencies for both $y, a|\hat{y}$ and $\hat{y}, a|y$ in Case (c), caused by the distinguished border between the two groups (see Fig. 1 (c)) is reflected by the extremely high Separation and Sufficiency scores. The value reported by NorMIX is slightly higher than 1, which can be attributed to numerical randomness in the learning process. It should be interpreted as a value close to 1.

Furthermore, Case (e) and (f) were designed to create distinct clusters in $y, \hat{y}$ scatter plots for different values of attribute a, resulting in little overlap between the two groups (see Fig. 2). NorMIX indeed detected these facts and reported high Independence scores for both cases, with a higher severity 0.8194 for Case (f) compared to 0.5889 for Case (e), indicating

Algorithm 1 NorMIX: **Nor**malized Mutual Information by Gaussian **MIX**tures as Regression Fairness Metrics

Require: A data set D consisting of target variable Y, model predictions $\hat{Y}$ and sensitive attributes A. Both Y and $\hat{Y}$ are continuous variables. And A is a categorical variable with non-negative integers $\{0, 1, 2, \ldots, C\}$, where C is a positive integer.

Require: K, positive integer number specifying the number of Gaussian components for Gaussian mixture learning.

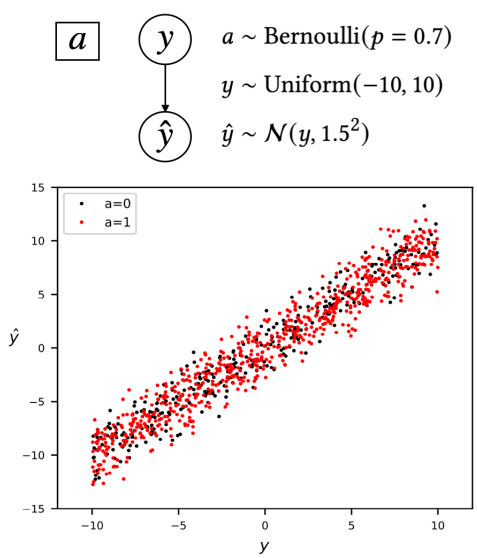
Require: Two functions: 1. *gmm_em*(d, K), a function to return a Gaussian mixture of K components from data d. It can learn one-dimensional GM or multi-dimensional GM based on the dimension of d; 2. *gmm_conditationl*$(\varphi(x_1, x_2), z)$ to return the parameters of the GM estimate of the conditional density of x_1 given x_2 if $z = 1$ or x_2 given x_1 if $z = 2$ from the joint GM $\varphi(x_1, x_2)$ based on Equations 14 - 17.

One-dimensional GM learning:
$\varphi(y) = \sum_{k=1}^{K} w_k \mathcal{N}(\mu_k, \mathcal{V}_k) \sim$ *gmm_em*$(D['Y'], K)$
$\varphi(\hat{y}) = \sum_{k=1}^{K} \hat{w}_k \mathcal{N}(\hat{\mu}_k, \hat{\mathcal{V}}_k) \sim$ *gmm_em*$(D['\hat{Y}'], K)$
# Learn GM given each unique value of A:
for $a \in \{0, 1, 2, \ldots, C\}$ **do**
    $\varphi(y|a) = \sum_{k=1}^{K} l_{ak} \mathcal{N}(u_{ak}, S_{ak}) \sim$ *gmm_em*$(D[A = a,' Y'], K)$
    $\varphi(\hat{y}|a) = \sum_{k=1}^{K} j_{ak} \mathcal{N}(\hat{u}_{ak}, \hat{S}_{ak}) \sim$ *gmm_em*$(D[A = a,' \hat{Y}'], K)$
end for
# Joint Two-dimensional GM learning and conditional GM derivation:
$\varphi(y, \hat{y}) = \sum_{k=1}^{K} r_k \mathcal{N}(U_k, \Sigma_k) \sim$ *gmm_em*$(D['Y',' \hat{Y}'], K)$
$\varphi(y|\hat{y}) = \sum_{k=1}^{K} g_k \mathcal{N}(h_k, Q_k) \sim$ *gmm_conditional*$(\varphi(y, \hat{y}), 1)$
$\varphi(\hat{y}|y) = \sum_{k=1}^{K} \hat{g}_k \mathcal{N}(\hat{h}_k, \hat{Q}_k) \sim$ *gmm_conditional*$(\varphi(y, \hat{y}), 2)$
for $a \in \{0, 1, 2, \ldots, C\}$ **do**
    $\varphi(y, \hat{y}|a) = \sum_{k=1}^{K} r_{ak} \mathcal{N}(U_{ak}, \Sigma_{ak}) \sim$ *gmm_em*$(D[A = 1, ['Y',' \hat{Y}']], K)$
    $\varphi(y|\hat{y}, a) = \sum_{k=1}^{K} g_{ak} \mathcal{N}(h_{ak}, Q_{ak}) \sim$ *gmm_conditional*$(\varphi(y, \hat{y}|a), 1)$
    $\varphi(\hat{y}|y, a) = \sum_{k=1}^{K} \hat{g}_{ak} \mathcal{N}(\hat{h}_{ak}, \hat{Q}_{ak}) \sim$ *gmm_conditional*$(\varphi(y, \hat{y}|a), 2)$
end for
# Independence score:
$\tilde{I}_{\textbf{Ind}} = \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|a^i)}{\varphi(\hat{y}_i)}}{-\sum_{a \in A}(\frac{n_a}{n} \ln \frac{n_a}{n})}$
Separation score:
$\tilde{I}_{\textbf{Sep}} = \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(\hat{y}_i|y_i, a^i)}{\varphi(\hat{y}_i|y_i)}}{-\frac{1}{n}\sum_{i=1}^{n} \ln(\frac{\varphi(y_i|a^i)P(a^i)}{\varphi(y_i)})}$
Sufficiency score:
$\tilde{I}_{\textbf{Suf}} = \frac{\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\varphi(y_i|\hat{y}_i, a^i)}{\varphi(y_i|\hat{y}_i)}}{-\frac{1}{n}\sum_{i=1}^{n} \ln(\frac{\varphi(\hat{y}_i|a^i)P(a^i)}{\varphi(\hat{y}_i)})}$
Return $\tilde{I}_{\textbf{Ind}}$, $\tilde{I}_{\textbf{Sep}}$, $\tilde{I}_{\textbf{Suf}}$

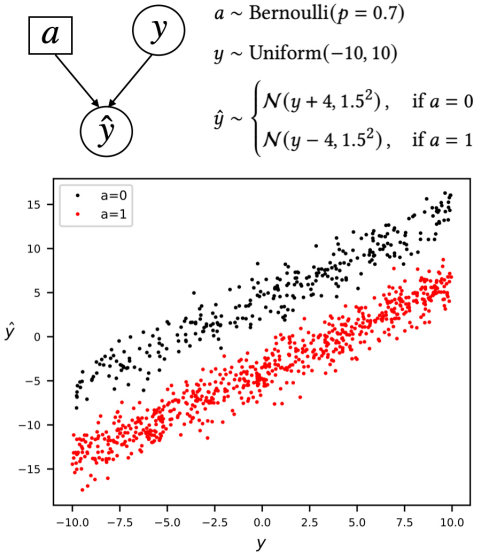
TABLE I
Fairness Score Comparison for Cases (a)-(d)

Case	Fairness Metric	Approx. True	NorMIX	Steinberg	Δ_1 [a]	Δ_2 [b]
(a)	$\tilde{I}_{\textbf{Ind}}$	0.0	0.0007	-0.003	0.00007	-0.003
	$\tilde{I}_{\textbf{Sep}}$	0.0	0.0021	-0.006	0.0021	-0.006
	$\tilde{I}_{\textbf{Suf}}$	0.0	-0.0013	-0.006	-0.0013	-0.006
(b)	$\tilde{I}_{\textbf{Ind}}$	0.3209	0.3279	0.271	0.007	-0.0499
	$\tilde{I}_{\textbf{Sep}}$	0.984	1.0059	0.89	0.0219	-0.094
	$\tilde{I}_{\textbf{Suf}}$	0.9765	1.0006	0.847	0.0241	-0.1295
(c)	$\tilde{I}_{\textbf{Ind}}$	0.0	0.0007	-0.015	0.0007	-0.015
	$\tilde{I}_{\textbf{Sep}}$	0.9843	1.0319	0.841	0.0476	-0.1433
	$\tilde{I}_{\textbf{Suf}}$	0.9907	1.0568	0.898	0.0661	-0.0927
(d)	$\tilde{I}_{\textbf{Ind}}$	0.0872	0.0873	0.082	0.0001	-0.0052
	$\tilde{I}_{\textbf{Sep}}$	0.3935	0.4276	0.324	0.0341	-0.0695
	$\tilde{I}_{\textbf{Suf}}$	0.3355	0.3728	0.258	0.0373	-0.0775
				Root Mean Square Error	0.0293	0.0755

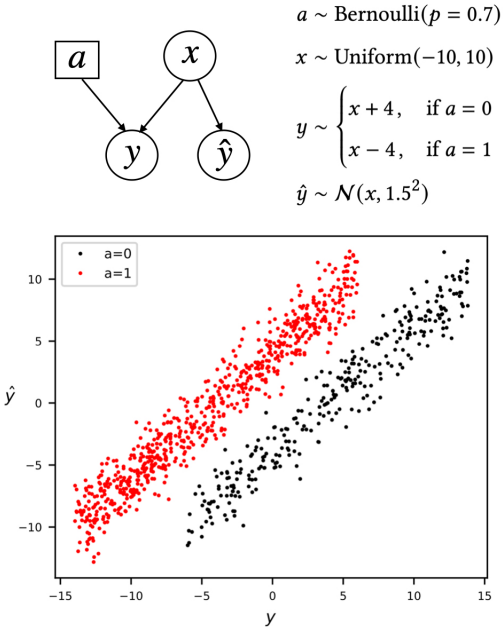
[a] Error term by NorMIX score − Approx. True
[b] Error term by Steinberg score − Approx. True



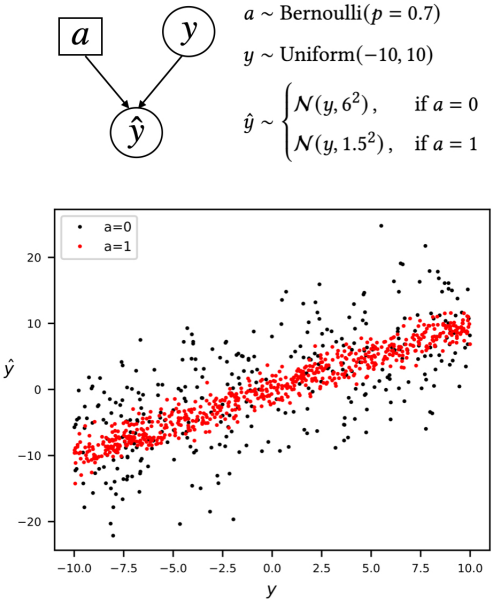
(a) Fair, $y, \hat{y}$ are independent of a.



(b) Biased, $\hat{y}$ is dependent on a (shift mean).



(c) Biased, $\hat{y}$ is independent of a, but becomes conditional dependent on a given y.

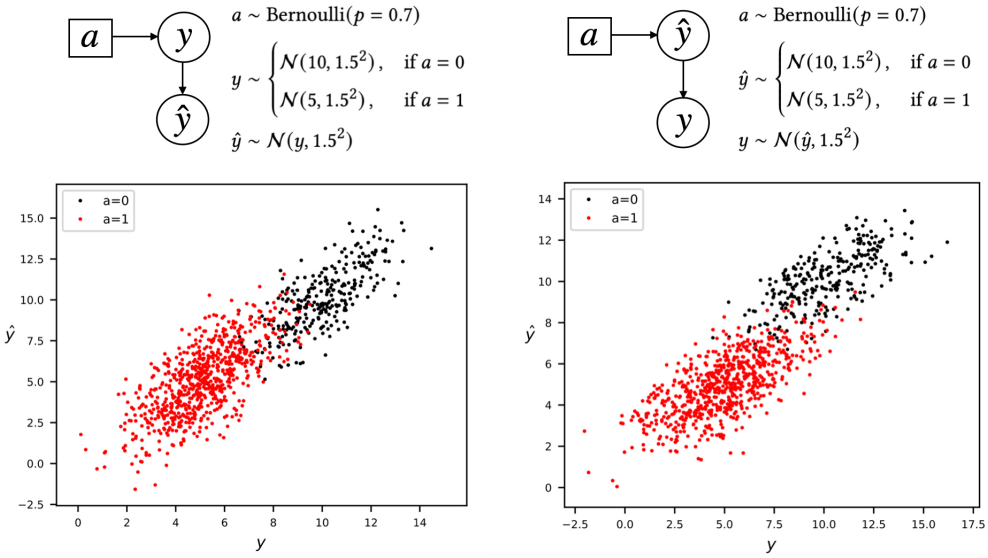


(d) Biased, $\hat{y}$ is dependent on a (shift variance).

Fig. 1. Simulation setting for Cases (a) - (d) with different fairness scenarios.

that $\hat{y}$ is more dependent on a in Case (f) than in Case (e). This is because $\hat{y}$ is directly impacted by a as designed in Case (f) (a is the direct parent of $\hat{y}$), whereas in Case (e), $\hat{y}$ is indirectly impacted by a through y. As previously noted, the near-zero scores reflect the absence of Separation in Case (e) and the absence of Sufficiency in Case (f).

In terms of complexity, NorMIX does not require training any classification models. Machine learning classifiers typically have many model choices and hyperparameters that need to be tuned. It is hard to choose and justify a classifier without knowing the true scores. By contrast, NorMIX directly focuses on density estimations in the continuous space without converting the original problem into discrete classifications. It learns GMMs from the joint space of $a, y, \hat{y}$ using the well-established EM algorithm. Please note that for fairness evaluation of regression models, the continuous joint space has only two dimensions: y and $\hat{y}$. This gives NorMIX an advantage to perform well on clustering without suffering of



(e) Biased, $\hat{y}$ is highly dependent on a.

(f) Biased, $\hat{y}$ is highly dependent on a.

Fig. 2. Simulation setting for demonstrating high bias but with zero Separation in Case (e) and zero Sufficiency in Case (f).

TABLE II
NorMIX Performance Results for Case (e) and (f)

Case	Fairness Metric	Approx. True	NorMIX	Δ^a
(e)	$\tilde{I}_{\text{Ind}}$	0.5889	0.5922	0.0033
	$\tilde{I}_{\text{Sep}}$	0.0	0.0066	0.0066
	$\tilde{I}_{\text{Suf}}$	0.5614	0.5726	0.0112
(f)	$\tilde{I}_{\text{Ind}}$	0.8194	0.8248	0.0054
	$\tilde{I}_{\text{Sep}}$	0.5608	0.5726	0.0046
	$\tilde{I}_{\text{Suf}}$	0.0	0.0066	0.0066
			Root Mean Square Error	0.0068

[a]Error term by NorMIX score − benchmark

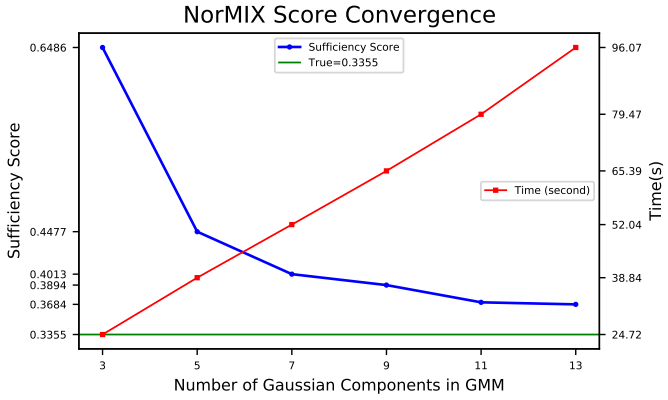


Fig. 3. Sufficiency Score Convergence for Case (d)

high dimensionality. Additionally, the only hyperparameter in GMM learning is the number of Gaussian components in the mixture. In general, increasing the number of Gaussian components improves the estimation accuracy of the GMM but also increases computational time. In our experiments, we observed that NorMIX computation time increases linearly with the number of Gaussian components. This is demonstrated by the example shown in Figure 3, where the Sufficiency score gradually approaches the true value as the number of Gaussians is increased from 3 to 13 for calculating the fairness scores in Case (d). Furthermore, it is worth noting that all empirical experiments for this paper were conducted in a pure Python environment, including the implementation of EM algorithm. This suggests that a distributed implementation of NorMIX should theoretically achieve even better computational performance.

B. Real data

For application purposes, we tested the NorMIX algorithm on a public dataset named Community and Crime dataset from [16]. This dataset contains counts of all reported violent crimes for $1,994$ communities across the United States. Each community has 128 features, including several demographic features from the census such as population density in percentage, average income and percentage of population that is unemployed. Following previous works, we defined communities where the percentage of black population is 50% or higher as the protected group 'black', while communities with less than 50% black population are categorized as the non-protected group 'other'. The community crime rate, a continuous value within the range of $[0, 1]$, serves as the target variable for this dataset.

We split the data into training (60%) and testing (40%) sets. The size of the training and testing set are $1,196$ and 798 respectively. Using XGBoost, we trained a regression model on the training data, and then made predictions on the testing data using the trained model. We then used the

true crime rates, model predictions, and the protected group definition mentioned earlier to calculate NorMIX scores for Independence, Separation, and Sufficiency. Table III shows the NorMIX scores obtained on the Community and Crime dataset.

TABLE III
NorMIX Fairness Scores for Community and Crime Dataset

$\tilde{I}_{\text{Ind}}$	$\tilde{I}_{\text{Sep}}$	$\tilde{I}_{\text{Suf}}$
0.4976	0.3709	0.0719

As shown in Fig. 4, the joint distributions of y and $\hat{y}$ vary significantly between different race groups. For the 'black' group, the distribution is mainly located on the high end with a larger spread and variance, whereas for the 'other' group, the distribution is more concentrated on the lower end with a tighter variance. Intuitively, we can determine that the Independence score for this XGBoost model will indicate bias (e.g., significantly higher than zero) because the model output is highly correlated with race. Moreover, for a given value of y (e.g., 0.4), we observe that the distribution of $\hat{y}$ varies significantly depending on race groups, with generally lower prediction rates for 'other' group and higher rates for 'black' group. This resulted in the moderately positive score for Separation (0.3709). On the other hand, for a given $\hat{y}$, the true rates from different race groups are less distinguishable horizontally, leading to a fair Sufficiency score (closer to zero). NorMIX reported a Sufficiency score of 0.0719, which is reasonable.

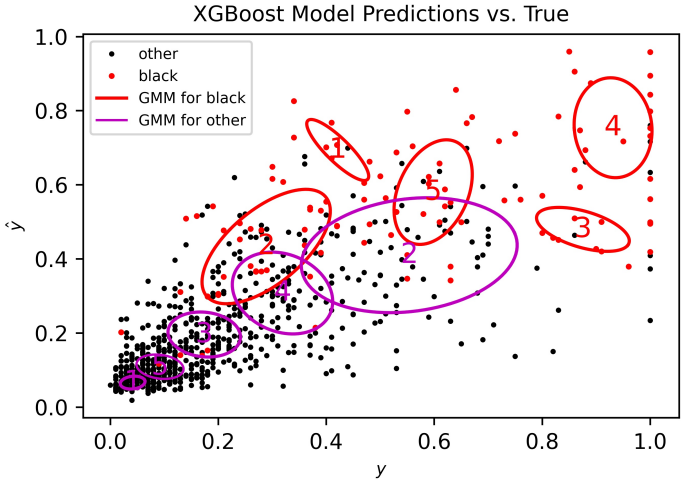


Fig. 4. Demonstration of 2-dimensional GMMs learned from XGBoost model predictions vs. the ground truth

V. Concluding Remarks and Future Work

Detecting fairness issues in regression models is a challenging and important topic, but there has been little work reported in the AI community over the past decade to quantify Independence, Separation, and Sufficiency in a regression setting. This lack of research has led to inconsistent approaches for addressing model bias in both classification and regression models. This paper introduces NorMIX, a novel algorithm that directly operates on conditional density estimation and computes fairness scores based strictly on the definitions of the three general fairness criteria. Unlike previous approaches that convert the problem into discrete formulations, simplify it with moment representations, or make other compromises, NorMIX calculates the fairness score at the density level by directly estimating the conditional densities themselves. NorMIX offers algorithmic flexibility, tunable accuracy, and easy integration of mature existing learning algorithms such as Expectation-Maximization (EM) algorithm for learning Gaussian mixture models from data, as well as analytical derivation from joint Gaussian mixture models to conditional Gaussian mixture models. Experiments conducted using both simulated and real datasets demonstrated the effectiveness of our approach. As shown in Section IV, the GMM density estimations were able to capture the underlying true distributions with reasonable accuracy. It is important to note that once the GMM for the 2-dimensional joint density is well learned, the conditional density of one continuous variable given another can be analytically derived and demonstrated to have good accuracy.

As mentioned earlier, in the context of evaluating fairness in regression models, the joint space for learning density is limited to two dimensions (y and $\hat{y}$). NorMIX takes advantage of this fact because the curse of dimensionality does not apply here. With a reasonable amount of data, we expect good estimates of the GMMs from this space. The number of Gaussian components is the only hyperparameter in GMM learning, and it plays a crucial role in determining the estimation accuracy and computation time. One caveat is that overfitting may occur if too many Gaussian components are used in learning GMMs. Exploratory data analysis can provide useful heuristics for determining a reasonable number of components to start with. Depending on the specific application, one can make a trade-off between accuracy and computational time by selecting an appropriate number of Gaussian components.

For future work, we recommend giving more attention to exploring the interpretation of normalized MI scores. Further, determining an appropriate fairness threshold is a domain-specific issue that requires practical guidance from subject matter experts. In terms of distributed implementation, we plan to develop NorMIX in the Spark environment [17] for potentially faster computations. Additionally, we intend to utilize NorMIX to assess bias in regression models. This can aid in evaluating mitigation methods and facilitating the investigation of potential bias mitigation techniques.

References

[1] Will Dobbie, Andres Liberman, Daniel Paravisini, and Vikram Pathania, "Measuring Bias in Consumer Lending," The Review of Economic Studies, Volume 88, Issue 6, November 2021, Pages 2799–2832. Available: https://doi.org/10.1093/restud/rdaa078

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. "Machine bias: There's software used across the country to predict future criminals, and it's biased against blacks," ProPublica, May, 23, 2016.

[3] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. "Fairness and Machine Learning: Limitations and Opportunities," fairmlbook.org. Available: http://www.fairmlbook.org.

[4] Alekh Agarwal, Miroslav Dudik, and Zhiwei S. Wu. 2019. "Fair Regression: Quantitative Definitions and Reduction-based Algorithms," In the Proc. of International Conference on Machine Learning 2019, pp.120–129.

[5] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. "A Convex Framework for Fair Regression," Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT (2017).

[4] Jack Fitzsimons, AbdulRahman Al Ali, Michael Osborne, and Stephen Roberts. 2019. "A general framework for fair regression," Entropy 21(8):741 (2019).

[5] Simon Caton and Christian Haas. "Fairness in Machine Learning: A Survey," Arxiv (2020). Available: https://arxiv.org/abs/2010.04053

[6] Daniel Steinberg, Alistair Reid, and Simon O'Callaghan. "Fairness Measures for Regression via Probabilistic Classification," CoRR (2020). Available: https://arxiv.org/abs/2001.06089

[7] Steffen Bickel, Michael Brückner, and Tobias Scheffer. "Discriminative Learning Under Covariate Shift," The Journal of Machine Learning Research 10 (2009), pp.2137–2155.

[8] Jing Qin. "Inferences for Case-Control and Semiparametric Two-Sample Density Ratio Models," Biometrika 85, 3 (1998), pp.619–630. Available: http://www.jstor.org/stable/2337391.

[9] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. "Density ratio estimation: A comprehensive review," RIMS Kokyuroku 1703 (2010), pp.10–31.

[10] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory* (second ed.), 2006, Wiley-Interscience.

[11] KC Chang and Wei Sun. "Scalable Fusion with Mixture Distributions in Sensor Networks," Proceedings of the 11th International Conference on Control Automation, Robotics and Vision (2010).

[12] Wei Sun, KC Chang, and Kathryn B. Laskey. 2010. "Scalable Inference for Hybrid Bayesian Networks with Full Density Estimations," proceedings of the 13th International Conference on Information Fusion (2010).

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, 2006, Springer.

[14] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*, 2006, MIT Press.

[15] Hsi Guang Sung. "Gaussian Mixture Regression and Classification," Rice University PhD Thesis (2004).

[16] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*, 2019. Available: http://archive.ics.uci.edu/ml

[17] Open source software. 2014. Apache Spark Official Website. Available: https://spark.apache.org/