Fairness-Aware Online Personalization

G Roshan Lal
LinkedIn Corporation
rlal@linkedin.com

Sahin Cem Geyik *
Facebook
scgeyik@fb.com

Krishnaram Kenthapadi *
Amazon AWS AI
kenthk@amazon.com

ABSTRACT

Decision making in crucial applications such as lending, hiring, and college admissions has witnessed increasing use of algorithmic models and techniques as a result of confluence of factors such as ubiquitous connectivity, ability to collect, aggregate, and process large amounts of fine-grained data using cloud computing, and ease of access to applying sophisticated machine learning models. Quite often, such applications are powered by search and recommendation systems, which in turn make use of personalized ranking algorithms. At the same time, there is increasing awareness about the ethical and legal challenges posed by the use of such data-driven systems. Researchers and practitioners from different disciplines have recently highlighted the potential for such systems to discriminate against certain population groups, due to biases in the datasets utilized for learning their underlying recommendation models.

We present a study of fairness in online personalization settings involving ranking of individuals. Starting from a *fair* warm-start machine learned model, we first demonstrate that online personalization can cause the model to learn to act in an unfair manner, if the user is biased in his/her responses. For this purpose, we construct a stylized model for generating training data with potentially biased features as well as potentially biased labels, and quantify the extent of bias that is learned by the model when the user responds in a biased manner as in many real-world scenarios. We then formulate the problem of learning personalized models under fairness constraints, and present a regularization based approach for mitigating biases in machine learning. We demonstrate the efficacy of our approach through extensive simulations with different parameter settings.

1 INTRODUCTION

Algorithmic models and techniques are being increasingly used as part of decision making in crucial applications such as lending, hiring, and college admissions due to factors such as ubiquitous connectivity, ability to collect, aggregate, and process large amounts of fine-grained data using cloud computing, and ease of access to applying sophisticated machine learning models. Quite often, such applications are powered by search and recommendation systems. In such applications, users may not always be able to articulate their information need in the form of a well-defined query, although they can usually determine if a displayed result (such as a potential candidate for hiring or college admission) is relevant to their information need or not. Hence, it may be desirable to personalize the results to each user based on their (explicit or implicit) responses, ideally in an online fashion. At the same time, human responses and decisions can quite often be influenced by conscious or unconscious biases, in addition to the relevancy of the results, causing the personalized models themselves to become biased. In fact, there is increasing awareness about the ethical and legal challenges posed by the use of such data-driven systems. Researchers and practitioners from different disciplines have highlighted the potential for such systems to discriminate against certain population groups, due to biases in the datasets utilized for learning their underlying recommendation models. Several studies have demonstrated that recommendations and predictions generated by a biased machine learning model can result in systematic discrimination and reduced visibility for already disadvantaged groups [3, 11, 17, 29]. A key reason is that machine learned models that are trained on data affected by societal biases may learn to act in accordance with them.

Motivated by the need for understanding and addressing algorithmic bias in personalized ranking mechanisms, we perform a study of fairness or lack thereof in online personalization settings involving ranking of individuals. Starting from a *fair* warm-start machine learned model[1], we first demonstrate that online personalization can cause the model to learn to act in an unfair manner, if the user is biased in his/her responses. For this purpose, we construct a stylized mathematical model for generating training data with potentially biased features as well as potentially biased labels. We quantify the extent of bias that is learned by the model when the user sometimes responds in a biased manner (either intentionally or unintentionally) as in many real-world settings. We study both bias and precision measures under different hyper-parameter choices for the underlying machine learning model. Our framework for generating training data for simulation purposes could be of broader use to evaluate fairness-aware algorithms in other settings as well. We then formulate the problem of learning personalized models under fairness constraints, and present a regularization based approach for mitigating biases in machine learning, specifically for linear models. We demonstrate the efficacy of our approach through extensive simulations with different parameter settings. To summarize, the contributions of our work are as follows:

- A stylized mathematical model for simulating the biased behavior of users and generating data that is amenable to learning a biased model, if labeled by such biased users.
- Demonstration of potential bias in online personalization systems for ranking/recommending individuals.
- A regularization based strategy to mitigate algorithmic bias during model training.

The rest of the paper is organized as follows. We first present a mathematical model for studying bias in online personalization systems and the associated experimental results (§2). Next, we propose a regularization based method to mitigate algorithmic bias and demonstrate its effectiveness through an empirical study (§3). Finally, we present the related work in §4 and conclude the paper in §5.

* Work done while the authors were at LinkedIn.

[1] We note that although such an unbiased model may not be available in practice, we show that the personalized model can become highly biased in spite of starting from an unbiased model.

2 STUDY OF FAIRNESS IN ONLINE PERSONALIZATION

We next present a study of fairness in online personalization settings involving ranking of individuals. Starting from a *fair* warm-start machine learned model, we first demonstrate that online personalization can cause the model to learn to act in an unfair manner, if the user is biased in his/her responses. For this purpose, we construct a stylized model for generating training data with potentially biased features as well as potentially biased labels, and quantify the extent of bias that is learned by the model when the user responds in a biased manner as in many real-world scenarios (§2.1). We then present experimental results from our simulation framework, where we consider both bias and precision measures under different hyper-parameter choices for the underlying machine learning model (§2.2 – §2.4). We chose to use a simulation framework since it allows us to model *biased users*, and further vary their levels of bias.

2.1 Data Model

We next present our stylized model for generating training data which captures some of the sources of unfairness that is usually present in real-world data, as listed in §A.2. We split our discussion into two parts. We describe how we generate feature vectors in §2.1.1 and we describe how we further label the data in §2.1.2.

2.1.1 Features. Suppose that our training data consists of n data points, where the i^{th} data point x_i is a vector consisting of m seemingly harmless attributes: $x_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,m})$. For the i^{th} data point, let a_i denote the corresponding value of the protected attribute. Of the m attribute values of x_i, let the first m_1 attribute values be independent of the protected attribute value a_i and the remaining m_2 attribute values be generated in a fashion dependent on a_i ($m = m_1 + m_2$). We can thus model the m attributes of each data point as being generated according to the scheme in Figure 1. As presented in the figure, for each i, we first generate the protected attribute a_i independently using a Bernoulli random variable which gives 1 with probability p_{group} and 0 with probability $1 - p_{group}$. We can interpret a_i as encoding the membership to some protected class like belonging to a particular gender, race, or age ≥ 40, with $a_i = 1$ as corresponding to the privileged group. For the other (non-protected) attributes, we choose those that are "harmless" (independent of the protected attribute, i.e., $x_{i,1}$ through x_{i,m_1}) to be drawn either from a uniform distribution between 0 and 1, or a normal distribution with their respective means and variances, independently. For the "proxy attributes" (dependent on the protected attribute, i.e., x_{i,m_1+1} through x_{i,m_1+m_2}), we draw them independently from normal distributions, whose means and variance are functions of the protected attribute value a_i. Although we describe the attribute values to be drawn from either a uniform distribution or a normal distribution for simplicity, we remark that our model can easily be extended to allow more general distributions, e.g., categorical distributions for categorical attributes.

Current Setting: While the above mathematical framework is quite general, in our experiments for the current work, we chose $m_1 = 1$ and $m_2 = 2$ so that $m = 3$, for simplicity. We therefore employed one harmless attribute which we draw uniformly in [0,1], and two proxy attributes which we draw from Normal distributions dependent on the protected attribute. Finally, we held the variance constant for all Normal distributions, although as discussed above, this does not necessarily have to be so.

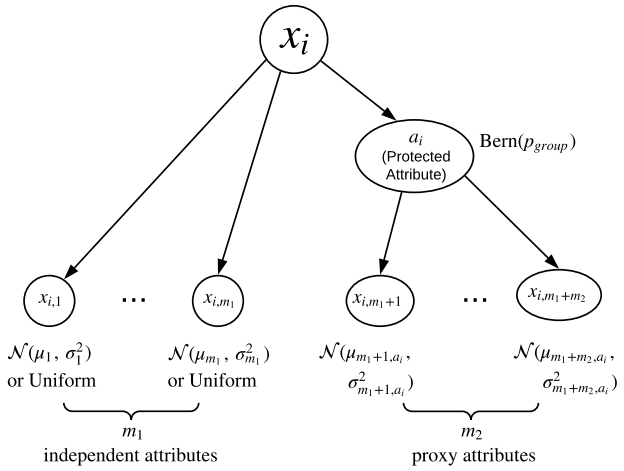


Figure 1: Model to simulate feature vectors of training data.

2.1.2 Labels. As stated earlier, our simulation framework enables us to model the behavior of *biased users* and vary their levels of bias, which we cannot obtain from real world datasets. Our labeling strategy for a potentially biased user is presented in Figure 2, and is as follows. Given the feature vector associated with a data point x_i, we need to label x_i as either "Accept" (1) or "Reject" (0). First, we determine whether the user who labels this data point would behave in a biased or unbiased manner, using an independent Bernoulli trial with parameter p_{bias}. This means that the user chooses to behave in a biased (unfair) manner with probability p_{bias}, and in an unbiased (fair) manner with probability $1 - p_{bias}$.

If the user chooses to be fair, he/she takes a fixed linear combination of the attribute values in the feature vector ($w_0 + \sum_{k=1}^{m} w_k x_{i,k}$, or in short, $w_0 + w^T x_i$) and labels the data point according to the sign of the linear combination (1 if the linear combination is non-negative, and 0 if it is negative). If the user chooses to be unfair, he/she labels the data point x_i purely based on the protected attribute value, a_i. More precisely, the user labels as "Reject" (0) whenever $a_i = 0$ and "Accept" (1) whenever $a_i = 1$.

The weights $[w_0, w_1, \ldots, w_m]$ are constants as defined by the user and unknown to the machine learning model which subsequently uses the data. The user chooses these weights without any knowledge of how the data is generated and has no knowledge of which attributes of the feature vector are independent of the protected attribute. Hence, the way the data is generated, the way it is labeled, and the way it is used for training the machine learning model are all blind to each other.

In the above model of simulating feature vectors and labeling them, we have captured the key aspects of unfairness listed in §A:

- *Proxy Attributes:* $x_{i,m_1+1}, \ldots, x_{i,m_1+m_2}$ serve as proxy attributes.

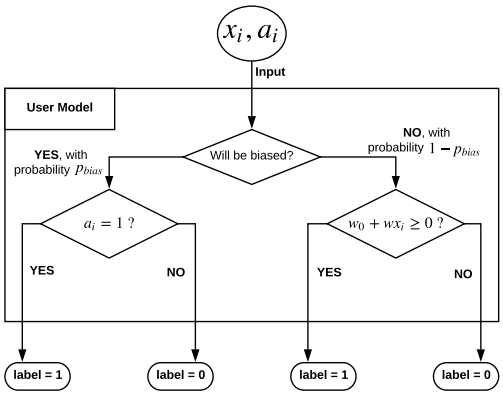


Figure 2: Model to label simulated feature vectors.

- *Quantity of training data for different values of the protected attribute:* This is controlled by the parameter p_{group} while generating the feature vectors.
- *Quality of training data for different values of the protected attribute:* This pertains to how the proxy attributes are generated from the protected attribute. Depending on the functions used for mapping the protected attribute value to the parameters of the distribution used for generating values of the proxy attributes, we may obtain training data with different quality for different groups. For example, the variance of the normal distributions used in our setting could be small for $a_i = 0$ and large for $a_i = 1$.
- *Noisy/biased labels in training data:* This is controlled by parameter p_{bias} which decides the user's level of unfairness. When the user behaves in an unfair manner, the resulting labels are biased, with $a_i = 0$ resulting in "Reject" (0) and $a_i = 1$ resulting in "Accept" (1).

We remark that our stylized training data generation model could be extended/generalized to incorporate a combination of "objectivity" and (possibly unconscious) biases, which may be closer to be more likely in practice. For instance, instead of considering the user to be either fully objective or fully biased when labeling, we could model the user as say, boosting the "objective" score based on the protected attribute value when determining whether to accept or reject. Another extension to our model would be to allow for the user to be partially biased: when the user is biased, we can view the label assigned by the user as a random variable dependent on the protected attribute value, and draw the value of the label from a Bernoulli distribution whose parameter could be influenced by the protected attribute value. For simplicity, we limit ourselves to the stylized model in this paper.

2.2 Experimental Setup

Next, we would like to describe our experimental setup to measure how different choices of hyper-parameters like learning rate and the number of training rounds affect fairness and precision in the outcomes of the online learning model.

Online Learning: In every iteration, we score all the data points in the training data and obtain the next data point for training by choosing the data point with the maximum score (highest possibility of being labelled as "Accept" as per the current model) that has not been used for training till now. The underlying setting is that we present one data point at a time to the user (we select the data point with the highest score, amongst those that have not been shown to the user so far), to which the user responds either positively or negatively. In either case, the online personalization model attempts to incorporate such user preference, so that a more personalized result can be presented next time to the user. We use the feature vectors and label of this new data point for training in the next round and this is repeated over and over again. The algorithm is started using a "warm-start" model in which data from a fair source is used to train the models by randomly sampling training data points from the fair dataset a number of times. We assume the availability of such a fair dataset so that we can then compare our final biased model against the warm-start baseline.

Our Setup: First, we generate 12000 data samples[2] at $p_{group} = 0.5$ and label them[3] in a fair manner (that is, p_{bias} is chosen to be 0 in §2.1.2, so that the labeling is based on the sign of the linear combination). We randomly sample 1000 data points from this set and use these data points to train a perceptron. We call this perceptron the **"warm-perceptron"** from here on. The data points that were generated are discarded beyond this point.

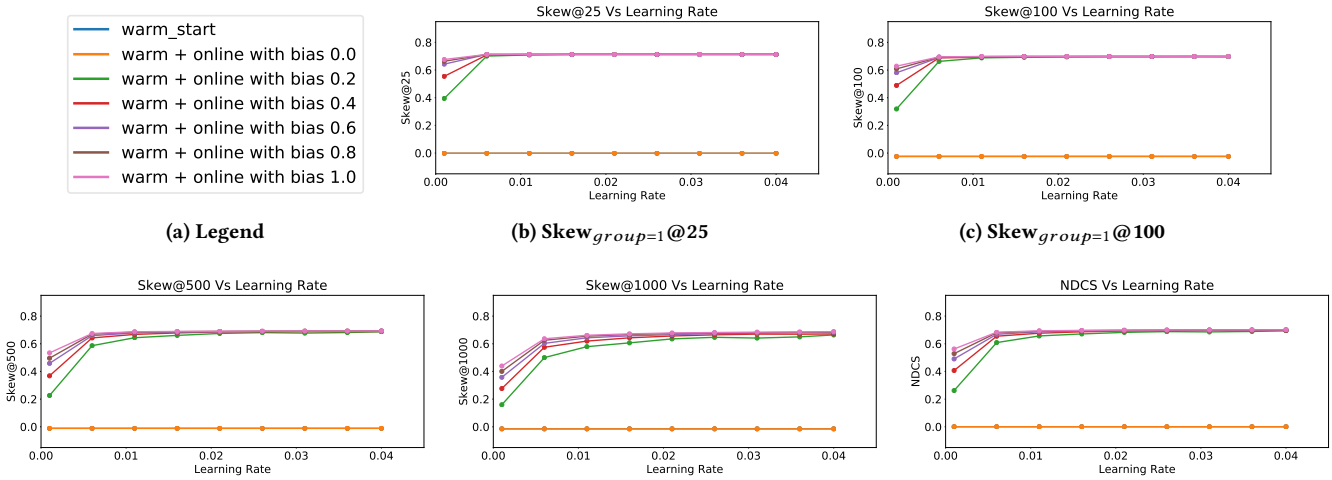
Next, we generate 12000 data samples at $p_{group} = 0.5$ and label them with the bias parameter set to a specific choice of p_{bias}. We start from the warm-perceptron model. In each round, we score all the 12000 data points using perceptron weights. We pick the point with highest score (the data point that is farthest from the perceptron on the positive side of the half-plane), obtain its label (following §2.1.2), and then update the warm-perceptron with this data point and its label.

Metrics: Our measures for evaluating bias are based on the assumption that the distribution of the protected attribute values for the top ranked candidates (i.e., based on the order with which a candidate gets to be shown to the user in our online personalization setting) should ideally reflect the corresponding distribution over the set of high quality or relevant candidates. A discussion on how this maps to equality of opportunity [19] for ranking problems is presented in [16].

Our first measure, Skew$_v$@k computes the logarithmic ratio of the proportion of candidates having v as the value of the protected attribute among the set of k highest scoring candidates to

[2] As discussed in §2.1.1, for our experiments, we utilize one "harmless" attribute ($x_{i,1}$ for each x_i), and two "proxy" attributes ($x_{i,2}$ and $x_{i,3}$ for each x_i). In our training data generation, for each data point, we choose the second and the third features to be dependent on the protected attribute value, but in opposite directions. For each i, if $a_i = 0$, we draw $x_{i,2}$ independently from the normal distribution with mean 0.35 and standard deviation 0.12, and if $a_i = 1$, we draw $x_{i,2}$ independently from the normal distribution with mean 0.65 and standard deviation 0.12. The mean values are reversed for the third attribute $x_{i,3}$. $x_{i,1}$ is drawn from the uniform distribution over [0,1].

[3] We utilized the following weights for the user label decision model (§2.1.2): $w_0 = -0.48$ (intercept), $w_1 = 0.35$, and $w_2 = w_3 = 0.28$. For both feature and label generation, we experimented with different choices of the parameters and observed qualitatively similar results.



(a) Legend **(b) Skew$_{group=1}$@25** **(c) Skew$_{group=1}$@100**

(d) Skew$_{group=1}$@500 **(e) Skew$_{group=1}$@1000** **(f) NDCS**

Figure 3: Skew and NDCS Results for the Final Model After Online Personalization Updates

the corresponding proportion among the set of relevant candidates:

$$\text{Skew}_v@\text{k} = \log_e\left(\frac{p_{v@k}}{p_{v,\text{qualified}}}\right) = \log_e\left(\frac{\frac{\sum_j \mathbb{1}(a_j=v \ \&\ \text{index}(j)<k)}{k}}{\frac{\sum_j \mathbb{1}(a_j=v \ \&\ w_0+w^T x_j \geq 0)}{\sum_j \mathbb{1}(w_0+w^T x_j \geq 0)}}\right), \tag{1}$$

where index(j) denotes the position of candidate j when ordered by decreasing scores, and $p_{v@k}$ and $p_{v,\text{qualified}}$ denote the fraction of candidates with protected attribute value v among the top-k recommended candidates and the overall qualified pool of candidates, respectively. Due to our simulation framework, we have further defined being "qualified" as being able to pass the linear criteria of the "fair" user ($w_0 + w^T x_j \geq 0$). This is one key advantage of utilizing the proposed mathematical framework for our experiments, since understanding which candidate is qualified is otherwise an extremely difficult task.

For each possible value v of the protected attribute, a negative skew corresponds to lesser representation of candidates with value v, while a positive skew corresponds to higher representation; a skew of zero is ideal.

Our second measure, $NDCS_v$ (Normalized Discounted Cumulative Skew) is a ranking version of the skew measure, inspired by *Normalized Discounted Cumulative Gain* [21] and the related bias quantification measures proposed in [35]. We define $NDCS_v$ as the cumulative skew over all values of k up to the total number of candidates shown, where each $Skew_v@k$ is weighted by a logarithmic discount factor based on k, and the sum is normalized by dividing by the sum of the discounting factors:

$$\text{NDCS}_v = \frac{1}{Z}\sum_j \frac{1}{\log_2(j+1)}\text{Skew}_v@\text{j}\,, \tag{2}$$

where,

$$Z = \sum_j \frac{1}{\log_2(j+1)}\,.$$

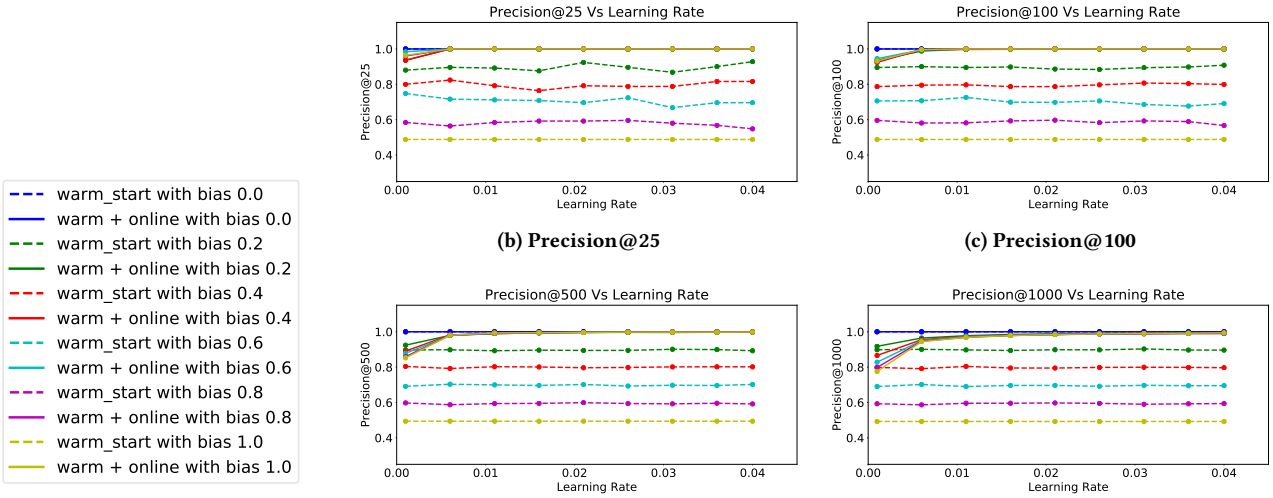
This measure is based on the intuition that achieving fairness in top results is more important than at the bottom, since the user is more likely to see and "Accept" (or even just be shown) the candidates at the top positions.

2.3 Evaluation of the Final Model After Online Personalization Updates

We first present the results of our study after a round of online personalization update scheme is applied. As discussed in §2.2 (see *Our Setup*), the initial model is what we call a warm-perceptron model, and we apply the online updates for 1000 candidates. At each new candidate, we choose the best candidate for the current model, get the feedback from the user (§2.1.2) and update the current model. At the end of 1000 rounds (candidates, feedback, and updates), we have generated a personalized model. In this section, we compare this personalized model (achieved by the end of 1000^{th} update) to the warm-perceptron model (which was the initial model before online updates), via an independent ranking of a large pool of candidates generated per §2.1, and checking the effect on the fairness and precision metrics.

Since the calculation of the fairness measures like $Skew@k$ (Eq. 1) and $NDCS$ (Eq. 2) requires the knowledge of the protected attribute distribution in the baseline population, we use the same data points used for online training and label them using a user model with $p_{bias} = 0$ and the same weights $[w_0, w_1, \ldots, w_m]$ for the purpose of calculating the baseline proportions (similar to the computation in Eq. 1). We use these labels from the fair user to get the baseline proportions (the binary protected attribute proportions in the qualified candidate set) of the two groups.

In Figure 3, we present Skew@k and NDCS results for the online personalization updates if we received the feedback to the recommended candidates from users with different p_{bias} values. We observe that as learning rate increases, the skew value increases at first and quickly saturates with increasing learning rate, for all types of users. From Figures 3b-3f, we can see this pattern across



(a) Legend **(b) Precision@25** **(c) Precision@100**

(d) Precision@500 **(e) Precision@1000**

Figure 4: Precision Results for the Final Model After Online Personalization Updates

all k for Skew@k and also in the NDCS metric. Furthermore, we see that the fairness metrics in *warm_start* (warm-perceptron) and *warm + online* (online personalization updates applied to warm-perceptron) with 0 bias are exactly the same. The reason is that the warm-perceptron model is also trained with data having 0 bias ($p_{bias} = 0$). Thus, the perceptron weights don't change further during the online rounds, since the weights are already learned in the warm-start phase. Hence, warm-start model and warm-start + online with 0 bias correspond to exactly the same perceptron weights. Furthermore, we observe that as the bias in training data increases, the Skew/NDCS metrics also increase. This can be explained as the perceptron learning the unfair behavior of the user by adjusting the weights of the attributes x_2 and x_3 which encode information about the sensitive attribute.

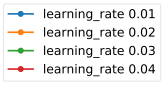
In Figure 4, we examine the precision of the warm-perceptron model vs. the subsequently online trained model on the biased user data at various p_{bias} values, as a function of the learning rate. We observe that as learning rate increases, the precision increases at first and quickly saturates with increasing learning rate. From Figures 4b-4e, we can see this pattern across all k for Precision@k (i.e., precision within the first k candidates recommended for users with different p_{bias}). Furthermore, we see that the warm-start model has very low precision while the online trained model has precision close to 1. This is because the warm-start model is trained on data labeled by a user with $p_{bias} = 0$ and is not precise enough to predict the response of users with large p_{bias} to the recommended candidates. The online model however learns the unfair behavior of user well enough for top k data points and hence precision@k is significantly improved. This means that online personalization, as expected, comes with a significant advantage for the "relevance" of recommendations, especially because it incorporated the algorithmic bias. This demonstrates the potential problem with applying such a preference understanding methodology, even when the initial model learned from a large pool of (say, unbiased) users is fair in recommendations.

2.4 Study of Personalized Model Evolution with Respect to Fairness and Precision

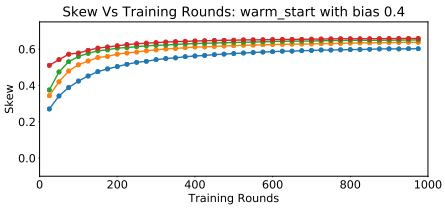
In this experiment, we interrupt the online learning process every 25 rounds and measure the fairness metrics for the ranked list of data points that has been shown thus far. Note that, semantically, this is different from the experiments presented in the previous section where we used the perceptron that is the outcome of the online updates over 1000 rounds (each round is presenting a candidate using the updated model, getting feedback, and further updating the model for the next round), and reranked all the 12000 data points to measure Skew@k, NDCS, and Prec@k. However, in this experiment, we interrupt the online learning process every 25 rounds and use the recommendations so far to measure the metrics. This can be seen as investigating the algorithmic bias that is being integrated into the online personalized model in real-time. Thus, this experiment measures the evolution of fairness in recommendations as we update the model with immediate feedback.

In Figure 5b, we evaluate the effect of learning rate and the number of recommendations on $\text{Skew}_{group=1}$@25 metric for personalizing towards the preference of a single user with $p_{bias} = 0.4$. Since the model learns the biased preferences of the user more and more with each subsequent training round, we observe that the skew increases with the number of training rounds. Furthermore, as the learning rate increases, the skew increases (keeping the number of rounds fixed), since we are moving the model "faster" towards the biased preferences of the user.

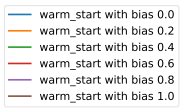
In Figure 5d, we look into the effect of p_{bias} and the number of recommendations on $\text{Skew}_{group=1}$@25 metric with a fixed learning rate of 0.01. We observe that the skew increases with the number of rounds (recommendations), as we learn the biased behavior of the user. Also, as p_{bias} increases, the skew grows faster with more number of rounds. For the model with $p_{bias} = 0$, the weights do not change much with more training rounds. Hence, the skew does not change significantly with the increase in the number of training



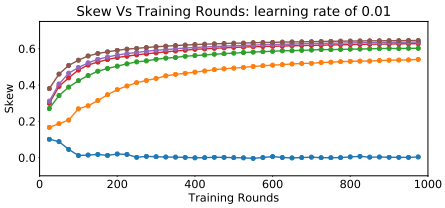
(a) Legend (fixed bias)



(b) Evolution with fixed bias



(c) Legend (fixed learning rate)



(d) Evolution with fixed learning rate

Figure 5: Evolution of Skew in Online Personalization

rounds. However, for users with high p_{bias}, the feature weights incorporate the algorithmic bias more and more as the number of training rounds increases and hence the skew increases. Overall, this study shows that for biased users, it is really easy to erase the unbiased/neutral nature of the warm-start model, if the goal is to understand and personalize the model to their (biased) preferences.

3 FAIR REGULARIZATION

We next discuss our method for restricting the learned models to be fair, followed by an evaluation of this approach in §3.2.

Consider the problem of constructing a fair machine learning model. Assume that we have data points in the form of (a feature vector, protected attribute, label) tuple, (x, a, y) which are generated from some distribution D. Suppose that we also have access to exactly N independent and identically distributed samples of training data sampled from D. Let us denote our training data as S $=\{ (x_1, a_1, y_1), (x_2, a_2, y_2), (x_3, a_3, y_3), \ldots, (x_N, a_N, y_N) \}$.

We are also given a space of hypotheses H, which are functions from the space of x to the space of y. We need to choose some hypothesis $h \in H$ using only the knowledge of the N data points in the training set S such that $h(x) = y$ with high probability and also h follows some chosen fairness criterion (for instance, the distribution of $h(x)$ and a are mutually independent), where (x, a, y) are sampled from the distribution D, conditioned on the fact that the training set S is given to us.

We restrict ourselves to the hypothesis space of linear functions from the space of x to y. In other words, for every $h \in H$, we can write $h(x) = w^T x$ where x is a feature vector. Furthermore, for our desired $h \in H$, we would like to impose the fairness condition of mutual independence of distributions of $h(x)$ and a when (x, a, y) are sampled from D, conditioned on the fact that the training set S is given to us.

We introduce matrices X and Y defined using training data as follows:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_N \end{bmatrix}$$

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & \ldots & a_N \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 & y_2 & y_3 & \ldots & y_N \end{bmatrix}$$

where x_i are column vectors (of features) a_i and y_i are scalar labels, for $i = 1, 2, 3, \ldots, N$. X, A, and Y are matrices completely formed out of training data S that is given to us and are not to be confused with $(x, a, y) \sim D$.

We assume that S is given to us and it can be used to select the $h \in H$ as per our desired constraints. Such a choice of h only depends on $S = \{(x_i, a_i, y_i) | i = 1, 2, 3, \ldots, N\}$ and that data points in the form of (x, a, y) can be sampled, independent and identically distributed from D and are especially independent of S. Such data points (x, a, y) would also be independent of the selected h which solely depends on S.

Now we present our fair learning problem for linear models as:

$$\textbf{Minimize}_w \; ||Y - w^T X||^2$$
$$\textbf{s.t.} \text{ the distributions of } w^T x \perp a$$
$$\text{when (x, a, y) is sampled from D given S.}$$

To make this problem more tractable, we make two relaxations.

Independence to Correlation: We first relax the constraint of $w^T x$ and a being independent to only being uncorrelated. It is well-known that independent variables are uncorrelated. Hence, zero correlation is a weaker constraint than independence. In the special case of jointly Gaussian variables, being uncorrelated implies independence. In the following equations, expectations are over $(x, a, y) \sim D$ given the training set S. w is dependent only on S and is independent of (x, a, y) by virtue of the iid sampling.

$$\mathbb{E}_{(x,a,y)\sim D|S}[w^T x a^T] - \mathbb{E}_{(x,a,y)\sim D|S}[w^T x]\mathbb{E}_{(x,a,y)\sim D|S}[a]^T = 0$$

$$\mathbb{E}[w^T x a^T] = w^T \mathbb{E}[x a^T] = w^T(\text{Cov}(x, a) + \mu_x \mu_a^T)$$

$$\mathbb{E}[w^T x]\mathbb{E}[a]^T = (w^T \mu_x)(\mu_a^T)$$

Therefore, $\mathbb{E}[w^T x a^T] - \mathbb{E}[w^T x]\mathbb{E}[a]^T = 0$ is equivalent to saying that $w^T \text{Cov}(x, a) = 0$ where $\mathbb{E}[x] = \mu_x$, $\mathbb{E}[a] = \mu_a$ and $\text{Cov}(x, a) = \mathbb{E}[(x - \mathbb{E}[x])(a - \mathbb{E}[a])^T]$.

Now, let us motivate towards our second relaxation. Suppose that we try to fit a linear model which predicts a_i from x_i for $i = 1, 2, 3, \ldots, N$ in training data. That is, we try to find the best fit least square solution to the equation $w_a^T X = A$ which is the same as the best fit solution to the equation $w_a^T (X - \bar{X}) = (A - \bar{A})$ where $\bar{X}$ and $\bar{A}$ are the respective sample means of X and A. Let us denote $\hat{X} = X - \bar{X}$ and $\hat{A} = A - \bar{A}$. Then the best fit least square solution is given by $\hat{X}\hat{X}^T w_a = \hat{X}\hat{A}^T$. Note that $\frac{1}{N}\hat{X}\hat{A}^T$ is the empirical estimate of $\text{Cov}(x, a)$ and $\frac{1}{N}\hat{X}\hat{X}^T$ is the empirical estimate of $\text{Cov}(x, x)$.

a to Linear Estimation of a: From the above discussion, we note that once we learn the best fit (in least squares sense) linear model to predict A from X, we can approximate $\text{Cov}(x, a) \approx \frac{1}{N}\hat{X}\hat{A}^T = \frac{1}{N}\hat{X}\hat{X}^T w_a \approx \text{Cov}(x, x) w_a$. Hence, we further relax the fairness constraint from $w^T \text{Cov}(x, a) = 0$ to $w^T \Sigma_x w_a = 0$ where $\text{Cov}(x, x) = \Sigma_x$.

With the above relaxations, learning a fair linear model reduces to solving:

$$\textbf{Minimize}_{w_a} \; ||A - w_a^T X||^2$$

$$\textbf{s.t.}\ ||w_a|| < \epsilon_a$$

and,

$$\textbf{Minimize}_w\ ||Y - w^T X||^2$$
$$\textbf{s.t.}\ ||w^T \Sigma_x w_a|| < \epsilon$$

where ϵ_a and ϵ are sufficiently small real numbers.

The above optimization problem has two parts. We first learn a best fit linear model from X to A. We use the weights learned in the first model to optimize for a fair best fit linear model from X to Y.

$$\textbf{Minimize}\ ||A - w_a^T X||^2$$
$$\textbf{s.t.}\ ||w_a|| < \epsilon_a$$

Note that the first minimization problem has an additional constraint of small L2 norm on weights w_a because the unconstrained optimization might be ill-posed. The contribution of the harmless attribute components in feature vectors' x_is towards the protected attribute is the ill-posed part. Hence, to make the contributions of the harmless attribute components in feature vectors x_is small, we impose the additional constraint of small L2 norm on the weights w_a.

The second optimization (for fair linear model) can be equivalently expressed as:

$$\textbf{Minimize}\ ||Y - w^T X||^2 + \lambda ||w^T w_{reg}||^2\ ,$$

where $w_{reg} = \Sigma_x w_a$. This is the Langrange Multiplier version or the regularized linear regression. We call this kind of regularization as Fair Regularization from now on. This can be solved exactly as:

$$(XX^T + \lambda w_{reg} w_{reg}^T)w = XY^T\ ,$$

or it can also be solved iteratively by a gradient update equation:

$$w_{n+1} = w_n + \eta(y_n - \text{sgn}(w_n^T x_n))x_n - \lambda(w_n^T w_{reg})w_{reg}\ .$$

Posing the fair learning problem as a regularized model has the benefit that:

- We need not transform the data to any kind of fair space. Fairness conditions are imposed on the model at train time, and not on training data which can still be used in its raw potentially unfair form.
- Fairness is imposed in an end-to-end single module compared to transforming the data to a fair space and then using the transformed data for the learning task at hand.

3.1 Extensions

Extending to second fairness condition: Let's first discuss how we can extend our approach for another kind of fairness constraint described in Section A.3. To achieve other kinds of fairness conditions like conditional independence ($C \perp A \mid Y$), our method can be tweaked by replacing the expectations with conditional expectations in the constraint of the second optimization problem. The resulting equation would result in $\Sigma_{x|y} = \mathbb{E}[XX^T \mid Y]$ in the place of $\Sigma_x = \mathbb{E}[XX^T]$. However, within the scope of our paper, we only work with the independence constraint.

Extending to more general hypothesis space: Finally, we would like to describe how our approach can be extended for more general hypothesis space. For general hypothesis space H, our first learning problem can be extended to learning the best hypothesis $h_a \in H$ which can predict a_i from x_i for $i = 1, 2, 3, \ldots, N$. Then in the second learning problem, we extend our regularization term which gets optimized with the training loss using the following insight:

$$w^T w_{reg} = w^T \Sigma_x w_a = \mathbb{E}[(w^T x)(w_a^T x)^T] = \mathbb{E}[(h(x))(h_a(x))^T]$$

Hence, while training to reduce the loss $L(h_n(x_n), y_n)$, we replace the loss with $L(h_n(x_n), y_n) + \lambda\ (h_n(x_n))(h_a(x_n)^T)^2$ where h_n is the hypothesis used and x_n is the training data point chosen in the n^{th} training iteration. Such a regularization has the effect that it penalizes the correlation of the model with another model that is the best fit that can be learned to predict the protected attribute. However, such a regularization is more noisy in general and takes more training rounds to converge compared to our case of linear model where the correlation can be better calculated and used at every iteration. Within the context of this paper, we only work with linear models, and leave such extensions to future work.

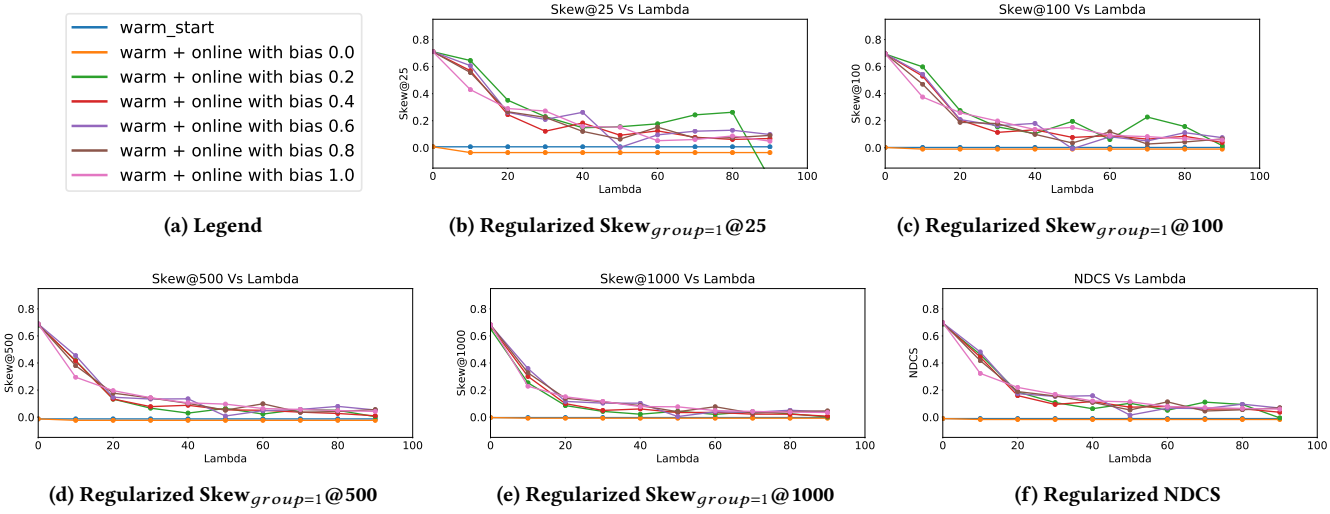
3.2 Evaluation of Mitigation using Fair Regularization

We next build upon our experiments in §2.3 with the fair regularization idea developed in §3. We study the effect of the regularization parameter λ on skew and precision. We follow the same setup as in §2.2 and §2.3 except that we fix a constant learning rate and experiment with different values of λ.

Figure 6 presents Skew@k and NDCS results as function of λ. As expected, Skew@k and NDCS drop towards 0 for all data sets generated with different amount of user bias as λ increases, which shows the efficacy of the fair regularization approach. However, we incur a trade-off in precision of the recommendations, since the regularization reduces the ability of the models to learn the biased preferences of the users. From Figure 7, we can see that as λ increases, Precision@k of the online updated model drops to Precision@k of the warm-perceptron model for each type of user with different p_{bias}. These results suggest an inherent tension between ensuring fairness and personalization. We could think of fair regularization also as attempting to address the over-fitting problem. Here, our goal is to prevent or minimize overfitting to the biased behavior of the user by imposing constraints on the learned weights for the proxy attributes. However, since the main objective of personalization is to learn and cater to the preferences of the given user as closely as possible (hence, for lack of a better word, "overfit" to the user's preferences, however biased they may be), regularization for ensuring fairness could hurt precision as we observed. The lesson learned is that we need to be careful when designing online personalized models and be aware of their potential to introduce algorithmic bias, even more prominently compared to offline models which may have been trained on data from a large set of users and tested for desired fairness properties.

4 RELATED WORK

Algorithmic bias, discrimination, and related topics have been studied across disciplines such as law, policy, and computer science [3, 17]. Two different notions of fairness have been explored in many recent studies: (1) *individual fairness*, which requires that similar people be treated similarly [11], and (2) *group fairness*, which requires that the disadvantaged group be treated similarly to the advantaged group or the entire population [28]. There is extensive



(a) Legend

(b) Regularized Skew$_{group=1}$@25

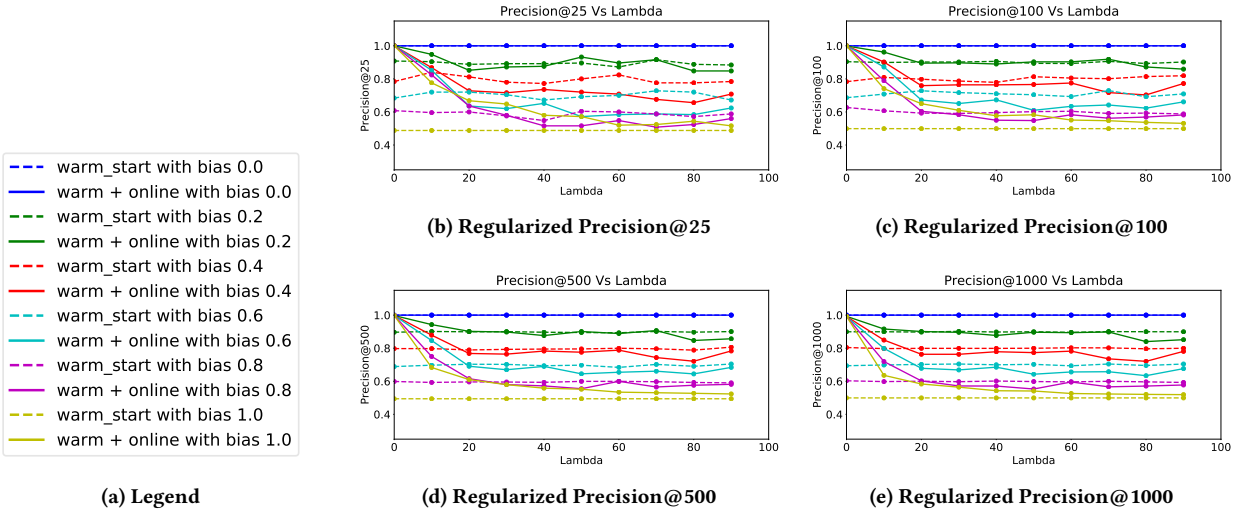
(c) Regularized Skew$_{group=1}$@100

(d) Regularized Skew$_{group=1}$@500

(e) Regularized Skew$_{group=1}$@1000

(f) Regularized NDCS

Figure 6: Regularized Skew and NDCS



(a) Legend

(b) Regularized Precision@25

(c) Regularized Precision@100

(d) Regularized Precision@500

(e) Regularized Precision@1000

Figure 7: Regularized Precision

work on identifying and measuring the extent of discrimination (e.g., [1, 8, 28]), on mitigation approaches in the form of fairness-aware algorithms (e.g., [7, 9–11, 14, 15, 18–20, 22, 23, 34, 36–38]), and on inherent trade-offs in achieving different notions of fairness [10, 11, 14, 23].

Modifying Model Training Algorithm for Fairness: While several studies have focused on data transformation/representation (pre-processing, post-processing, or during training) for ensuring fairness (e.g., [12, 13, 24, 38]), there has also been work on modifying the model training algorithm itself. Regularization based methods for fairness have been proposed in [22, 25, 30]. To meet the independence constraint of fairness, an approximate estimate of mutual information of the classifier output and the protected attribute is computed and used as a regularizer in [22]. This approach is extended in [25] for continuous protected attributes. Correlation between the protected attribute and the classifier output is further used as the regularizer in [30]. While we use a similar correlation based approach for performing classification, our method differs in the following key aspects. We first train an auxiliary model for predicting the protected attribute in terms of other features, and then guide our main model to be uncorrelated with this auxiliary model. Our problem setting is also slightly different. We use online learning wherein the training samples are not picked uniformly at random, but in a special way where we only show the highest scored candidate to the user for feedback in each iteration. The idea of auxiliary model has earlier been used in the context of training an adversarial neural network in a fair manner [4], but our training method is significantly different. While the auxiliary model is trained to underperform in [4], we train the auxiliary model

to perform well and use it to steer the main model away from it. Another recent paper [31] penalizes the covariance between the probability assignment of a user to its correct label, and his/her name's embedding, without access to protected attributes.

Some recent work [5, 27] have also dealt with achieving fairness via regularization utilizing pair-wise comparisons where paired items are chosen from different protected attribute groups. Both of these works in general attempt learn models which have similar probability to assign a larger score to better items than a worse item, regardless of the protected attribute.

Finally, a meta-learning approach through regularization is proposed in [33], where the authors propose to learn a model that performs well and fair across a set of tasks, and use a small size sample for each task. Within each task they apply a regularization term towards getting the fair model which is based on the average probability that a class is to be classified as positive given the protected attribute (parity), or given the protected attribute and label (equalized odds or equal opportunity).

Fairness in Ranking: There has been a lot of recent work on fairness in ranking [2, 6, 9, 26, 32, 35, 37]. Algorithms for reranking subject to fairness constraints, specified as the maximum (or minimum) number of elements of each class that can appear at any position in the ranking, have been proposed in [9, 37]. Fairness measures for ranking have been proposed in [35]. Fairness constraints are modeled as linear constraints along with a optimizing for relevance in [9]. Fairness in ranking and feedback from it are modeled as a stochastic bandit problem in [32]. Achieving fairness in a dynamic learning to rank setting using a fair controller is discussed in [26].

5 CONCLUSION

Considering the importance of understanding and addressing algorithmic bias in personalized ranking mechanisms, we formulated the problem of studying fairness in online personalization. We performed an investigation of potential biases possible in online personalization settings involving ranking of individuals. Starting from a fair warm-start machine learned model, we empirically showed that online personalization can cause the model to learn to act in an unfair manner, if the user is biased in his/her responses. As a part of this study, we presented a stylized mathematical model for generating training data with potentially biased features as well as potentially biased labels, and quantified the extent of bias that is learned by the model when the user sometimes responds in a biased manner as in plausible real-world settings. The ideas/intuition underlying our framework for generating training data for simulation purposes are likely to be of broader interest to the research community for evaluating fairness-aware algorithms in other scenarios, where either suitable datasets may not be available or it may be infeasible to perform extensive experimentation with real datasets. We then formulated the problem of learning personalized models under fairness constraints, and proposed a regularization based approach for mitigating biases in machine learning, specifically for linear models. We evaluated our methodology through extensive simulations with different parameter settings.

ACKNOWLEDGMENTS

The authors would like to thank other members of LinkedIn Careers and Talent Solutions teams for their collaboration for performing our experiments, and Deepak Agarwal, Erik Buchanan, Patrick Cheung, Patrick Driscoll, Nadia Fawaz, Joshua Hartman, Rachel Kumar, Ram Swaminathan, Hinkmond Wong, Ryan Wu, Lin Yang, Liang Zhang, and Yani Zhang for insightful feedback and discussions.

REFERENCES

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica*, 2016.
[2] A. Asudeh, H. V. Jagadish, J. Stoyanovich, and G. Das. Designing fair ranking schemes. *arXiv:1712.09752*, 2017.
[3] S. Barocas and M. Hardt. Fairness in machine learning. In *NIPS Tutorial*, 2017.
[4] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. In *FAT–ML Workshop*, 2017.
[5] A. Beutel et al. Fairness in recommendation ranking through pairwise comparisons. In *KDD*, 2019.
[6] A. J. Biega, K. P. Gummadi, and G. Weikum. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*, 2018.
[7] T. Calders and S. Verwer. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2), 2010.
[8] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 2017.
[9] L. E. Celis and N. K. Vishnoi. Fair personalization. In *FATML*, 2017.
[10] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *KDD*, 2017.
[11] C. Dwork, M. Hardt, T. Pitassi, and R. Z. Omer Reingold. Fairness through awareness. In *ITCS*, 2012.
[12] H. Edwards and A. Storkey. Censoring representations with an adversary. In *ICLR*, 2016.
[13] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *KDD*, 2015.
[14] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian. On the (im) possibility of fairness. *arXiv:1609.07236*, 2016.
[15] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth. A comparative study of fairness-enhancing interventions in machine learning. *arXiv:1802.04422*, 2018.
[16] S. C. Geyik, S. Ambler, and K. Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to LinkedIn talent search. In *KDD*, 2019.
[17] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *KDD Tutorial on Algorithmic Bias*, 2016.
[18] S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE TKDE*, 25(7), 2013.
[19] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NIPS*, 2016.
[20] S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in reinforcement learning. In *ICML*, 2017.
[21] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. on Information Systems (TOIS)*, 2002.
[22] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *ECML PKDD*, 2012.
[23] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *ITCS*, 2017.
[24] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel. The variational fair autoencoder. In *ICLR*, 2016.
[25] J. Mary, C. Calauzenes, and N. E. Karoui. Fairness-aware learning for continuous attributes and treatments. In *ICML*, 2019.
[26] M. Morik, A. Singh, J. Hong, and T. Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*, 2020.
[27] H. Narasimhan, A. Cotter, M. Gupta, and S. Wang. Pairwise fairness for ranking and regression. In *AAAI*, 2020.
[28] D. Pedreschi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *KDD*, 2008.
[29] D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *SDM*, 2009.
[30] A. Pérez-Suay, V. Laparra, G. Mateo-García, J. Muñoz-Marí, L. Gómez-Chova, and G. Camps-Valls. Fair kernel learning. In *ECML PKDD*, 2017.
[31] A. Romanov et al. What's in a name? reducing bias in bios without access to protected attributes. In *NAACL*, 2019.
[32] A. Singh and T. Joachims. Fairness of exposure in rankings. In *KDD*, 2018.

[33] D. Slack, S. A. Friedler, and E. Givental. Fairness warnings and fair-MAML: learning fairly with minimal data. In *FAccT*, 2020.
[34] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro. Learning non-discriminatory predictors. In *COLT*, 2017.
[35] K. Yang and J. Stoyanovich. Measuring fairness in ranked outputs. In *SSDBM*, 2017.
[36] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, 2017.
[37] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. FA*IR: A fair top-k ranking algorithm. In *CIKM*, 2017.
[38] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *ICML*, 2013.

A APPENDIX

A.1 Legal Frameworks on Bias and Discrimination

Over the last several decades, legal frameworks around the world have prohibited the use of sensitive attributes such as race, gender, sexual orientation, and age for different forms of decision making. We refer to such attributes whose use in certain decision making has been prohibited by law as *protected attributes*. Examples of attributes declared as protected by law in the United States include:

- Race, Color, Sex, Religion, National origin (Civil Rights Act of 1964; Equal Pay Act of 1963),
- Age (Age Discrimination in Employment Act of 1967),
- Disability status (Rehabilitation Act of 1973; Americans with Disabilities Act of 1990), and
- Veteran status (Vietnam Era Veterans' Readjustment Assistance Act of 1974).

When legal frameworks prohibit the use of such protected attributes in decision making, there are usually two competing approaches on how this is enforced in practice: *Disparate Treatment* vs. *Disparate Impact*. Avoiding disparate treatment requires that such attributes should not be actively used as a criterion for decision making and no group of people should be discriminated against because of their membership in some protected group. Avoiding disparate impact requires that the end result of any decision making should result in equal opportunities for members of all protected groups irrespective of how the decision is made. This can often mean that certain forms of affirmative action based on protected attributes may be needed to encourage or improve opportunities for certain protected groups to offset the biases present in the society due to historical reasons. Our work can be thought of as following the approach of avoiding disparate impact.

A.2 Sources of Bias

To prevent a machine learning model from incorporating biases, it is not enough to prohibit the usage of protected attributes. Often one or more protected attributes could be correlated with seemingly harmless attributes (hereafter denoted as "proxy attributes"). Presence of proxy attributes in a machine learning model potentially allows it to extract information on protected attributes and further use it to make decisions in an unfair manner.

Some possible reasons for biases in the outcomes of machine learning models include:

- Presence of proxy attributes in the set of features utilized for training,

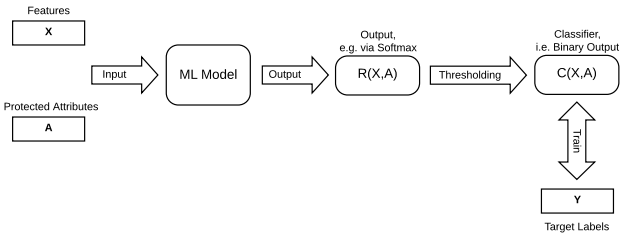


Figure 8: Formal Machine Learning Model

- Different quality and quantity of samples from different protected groups in training data, and
- Structured noisy labels in training data with different amounts of noise depending upon protected attributes.

In §2.1, we describe our stylized model of simulating training data which embodies most of the above sources of algorithmic bias.

A.3 Different Notions of Fairness

Consider the formal setup of a machine learning model shown in Figure 8. Let X be the feature vector that is used by the machine learning model as input and A be the corresponding protected attribute, which, for simplicity, is assumed to be a binary variable taking values 1 and 0 (e.g., whether sex = 'female'; whether age ≥ 40). Let $R(X, A)$ be the output of the model, which can be thought of as the result of a softmax procedure, and in the range $(0, 1)$. A thresholding scheme is applied on the output $R(X, A)$ to convert it into binary labels $C(X, A)$. The model is trained such that $C(X, A)$ follows the binary labels Y in training data. In such a setting, we could consider multiple approaches to formalize what we mean by fairness, leading to the following definitions [3].

(1) $C \perp A$: Independence condition requires that output is independent of the protected attribute. This is often called statistical parity where chances of "Accept" (or "Reject") conditioned on the value of the protected attribute are the same for both protected groups, i.e.,

$$P(C = 1 \mid A = 1) = P(C = 1 \mid A = 0) = P(C = 1).$$

Many approximate versions of this are often recommended by guidelines such as the Uniform Guidelines for Employee Selection Procedures, adopted by certain U.S. government agencies in 1978. According to these guidelines, adverse or disparate impact is determined by using the "four-fifths or 80% rule" which requires that the selection rate for any race, sex, or ethnic group be at least four-fifths (or 80%) of the rate for the group with the highest rate for the process to be considered as not having adverse impact. For the case of binary protected attribute, this corresponds to

$$\frac{1}{0.8} \geq \frac{P(C = 1 \mid A = 1)}{P(C = 1 \mid A = 0)} \geq 0.8.$$

A potential problem with this criterion is that often training data might not support equal "Accept" chances for all protected groups and hence enforcing such an independence constraint might hurt the precision or other performance measures of the algorithm.

(2) $C \perp A \mid Y$: Conditional independence condition requires that the model makes equal amounts of mis-classifications for all protected groups, i.e.,

$$P_{A=1}(C = 1 \mid Y = 1) = P_{A=0}(C = 1 \mid Y = 1),$$

$$P_{A=1}(C = 1 \mid Y = 0) = P_{A=0}(C = 1 \mid Y = 0).$$

Thus, it translates to enforcing equal true positive rates and false positive rates for all protected groups. This criterion can allow the model training to proceed to fit to the data with high precision.

(3) $Y \perp A \mid C$: This condition is similar to the previous condition, but with the roles of C and Y reversed. Note that C is trained to follow Y in training data and hence it makes sense to treat them equivalently.

In this paper, we focus on the independence condition, i.e., (1) above, based on which we devised an approach to mitigate bias in §3.