Achieving Outcome Fairness in Machine Learning Models for Social Decision Problems

Boli Fang[1] , **Miao Jiang**[2] , **Pei-yi Cheng**[1] , **Jerry Shen**[3] and **Yi Fang**[4]
[1]Dept of Computer Science, Indiana University
[2]Dept of Intelligent Systems Engineering, Indiana University
[3]Sol Price School of Public Policy, University of Southern California
[4]Dept of Computer Science and Engineering, Santa Clara University
{bfang, miajiang,peicheng}@iu.edu, haoxuans@usc.edu, yfang@scu.edu

Abstract

Effective complements to human judgment, artificial intelligence techniques have started to aid human decisions in complicated social decision problems across the world. Automated machine learning/deep learning(ML/DL) classification models, through quantitative modeling, have the potential to improve upon human decisions in a wide range of decision problems on social resource allocation such as Medicaid and Supplemental Nutrition Assistance Program(SNAP, commonly referred to as Food Stamps). However, given the limitations in ML/DL model design, these algorithms may fail to leverage various factors for decision making, resulting in improper decisions that allocate resources to individuals who may not be in the most need of such resource. In view of such an issue, we propose in this paper the strategy of *fairgroups*, based on the legal doctrine of *disparate impact*, to improve fairness in prediction outcomes. Experiments on various datasets demonstrate that our fairgroup construction method effectively boosts the fairness in automated decision making, while maintaining high prediction accuracy. [1]

1 Introduction

As defined by the United Nations Sustainable Development Goals, equality, fairness, and sustainability are top priorities for developed and developing nations across the world when social decision problems arise [Nations, 2016]. In particular, social decision problems, such as the proper allocation of strategic social resources including medical and food subsidies, are vital for the well-being of citizens across different countries in the world. For instance, according to the American Community Survey [Bureau, 2017], millions of American households are regularly receiving governmental assistance in receiving Medicaid and SNAP, compensation schemes designated for low-income individuals to receive proper reimbursement for necessary medical treatment and basic food access respectively. These programs have been instrumental in protecting the interests of citizens across the social spectrum.

Despite the enormous scale of these federal subsidy programs, proper distribution of the financial resources remains an issue that cannot be ignored, since subjective resource allocation based on human-expert judgments often leads to inaccuracies that negatively impact the decisions being made. It is noted in [Bureau, 2017], for example, that a substantial portion of these poor households are not yet receiving Medicaid for a variety of reasons. On the other hand, out of the households that are receiving Medicaid, a highly non-trivial amount - around 56% - of these households do not live under poverty [Bureau, 2017], and similar issues can be observed in SNAP allocation [Bureau, 2017]. Such disparity and inequality across the US behoove decision makers to introduce policies that better take various factors into consideration, and recent advancements in machine learning and deep learning algorithms have offered objective insights into similar problems in social policy enactment [Morse, 2018].

However, given the limitations of ML/DL algorithms and the bias in parameter choices and selection, the issue of fairness has also been the focus for a lot of current machine learning research[Zadrozny, 2004]. Depending on the nature of the resource allocation problem, one can group the factors/features of data into two categories: *protected* factors/features which are of priority in determining fairness, and *unprotected* factors/features which do not carry as much priority in decision making. In the context of Medicaid eligibility, for example, poverty level is the most prominent protected feature since the main purpose of Medicaid is to serve the low-income sector of society. It is important, therefore, to include as many individuals living under poverty into the program as possible, while minimizing the number of individuals that do not need such assistance so as to allow for the optimal allocation of the finite monetary and health resources.

Thus, given such considerations, we introduce in this paper a novel algorithm centered on the notion of *fairgroups* to make fair decisions in social decision problems, while maintaining a high degree of prediction accuracy. Here, the notion of fairness is based on the legal doctrine of *disparate impact* [Feldman *et al.*, 2015], which calls for similar levels of representation for all the groups of people in different decision

[1] The source code of our experiments is available at https://github.com/miaojiang1987/AI-for-fairness.

outcome classes. Our contributions in this work can be summarized as follows:

1. We provide an outcome-fairness algorithm for social decision problems by constructing *fairgroups*, which help achieve fairness with respect to protected features.
2. Our algorithm also takes into consideration unprotected features while making decisions on fairness, so that the overall classification accuracy remains high.
3. Our introduced method to achieve fairness is easily adaptable to other decision making problems involving the distribution of scarce resources.

2 Related Work

Previous work on fairness in machine learning can be largely divided into two groups. The first group has centered on the mathematical definition and existence of fairness. Along this track, alternative measures such as statistical parity, disparate impact, and individual fairness have been produced in [Chierichetti *et al.*, 2017], [Kamishima *et al.*, 2011]. [Kleinberg *et al.*, 2016] suggests that including ”protected” features in algorithms would increase the equity and efficiency of models. Moreover, [Corbett-Davies and Goel, 2018] points out that currently popular measures of fairness suffer from problematic statistical limitations or even lead to negatively impact on the group that researchers intend to protect.

The second group has centered on algorithms to achieve fairness. Along the route of disparate impact, [Chierichetti *et al.*, 2017] introduces the notion of *protected and unprotected features* which will be used in our paper. The notion of disparate impact has found its way to a wide range of machine learning problems such as k-clustering [Rösner and Schmidt, 2018; Schmidt *et al.*, 2018]. For examples of other applications, [Zafar *et al.*, 2015] [Joseph *et al.*, 2016] builds fair classifier leveraging both maximizing accuracy subject to fairness constraints and maximizing fairness subject to accuracy constraints so that the measure can ensure disparate treatment and disparate impact. [Zink and Rose, 2019] adds fairness considerations to the regression objective function that predict continuous rather than binary outcomes.

3 Fairness Model

In this section we present a novel strategy by constructing *fair-groups* to achieve fairness in machine learning models. This strategy adopts the notion of fairness as related to *disparate impact* [Feldman *et al.*, 2015], where practices based on neutral rules and laws may still more adversely affect individuals with one protected feature than those without.

3.1 Problem Formulation

We first define the terminology to be used in subsequent description. A *protected feature* is a feature that carries special importance and is of priority when making relevant decisions. An *unprotected feature*, on the other hand, is of relative minor importance in decision making. Since the problem in our paper primarily focuses on discrete label classification with discrete features, we assume that the protected traits and label classes are both binary. Given a protected feature A along with the dataset, the *balance* B of the dataset with respect to A is defined as

$$Bal(A) = \min\{\frac{\#\{A=0\}}{\#\{A=1\}}, \frac{\#\{A=1\}}{\#\{A=0\}}\} \in [0,1],$$

where $Bal(A) = 0$ refers to the case of all data points having the same feature value of A, and $Bal(A) = 1$ refers to the case where $\#\{A=0\} = \#\{A=1\}$. Intuitively, the Balance reflects the ratio between the minority and the majority.

Intuitively, by the definition of fairness with respect to disparate impact [Chierichetti *et al.*, 2017], a dataset is fair only when the ratio between the minority and the majority(with respect to certain attributes/features in discussion) is not too small, ideally above a certain threshold α. We therefore call a dataset α-*fair* with respect to feature A if the balance of A does not go below a certain number $\alpha \in [0,1]$. In other words, a dataset is α-disparate with respect to A if the groups with 2 different values in A have a bounded and relative balanced numerical ratio between $\frac{1}{\alpha}$ and α. We also say that a classification is (α, i)-fair if the group corresponding to label i in the classification class $L = \{+,-\}$ is α-fair, meaning that the protected feature is fairly represented with balance at least α in group i. Our goal in this paper is to develop an algorithm that produces (α, i)-fair classification with respect to label i.

3.2 Fairgroup Construction

We provide in this section the details of the algorithms we will use to achieve fairness in machine learning model. Here in our paper our focus lies on regression and decision tree algorithms.

Assume that we already have a machine learning algorithm C which yields predictions for data points. In most cases, C does not yield α-fair classification outcome. Overall, our algorithm computes the feature importance scores and use them to quantify the similarities between different data points. It then greedily constructs fair-groups satisfying user-defined balance constraints. Finally, the algorithm conducts classification on the data points with C while taking the properties of the fairgroups into consideration.

Feature Importance Computation

Most of the social decision problems involve different features of varying degrees of relevance and importance to the goal. Therefore, we need a measure to describe the similarity between these features. Depending on the nature of the machine learning model in discussion, we consider two different methods to compute feature importance scores. For regression models, a natural choice of such a measure would be the correlation between X_i and Y, since it provides quantitative information about the statistical relationship between X_i and Y and the strength of their correlation. For each feature X_i, we compute the correlation coefficient between X_i and the outcome decision vector Y to determine the positive/negative contributions of each feature X_i to the final classification outcome:

$$Corr(X_i, Y) = \frac{E[X_iY] - E[X_i]E[Y]}{\sqrt{Var(X_i)Var(Y)}}.$$

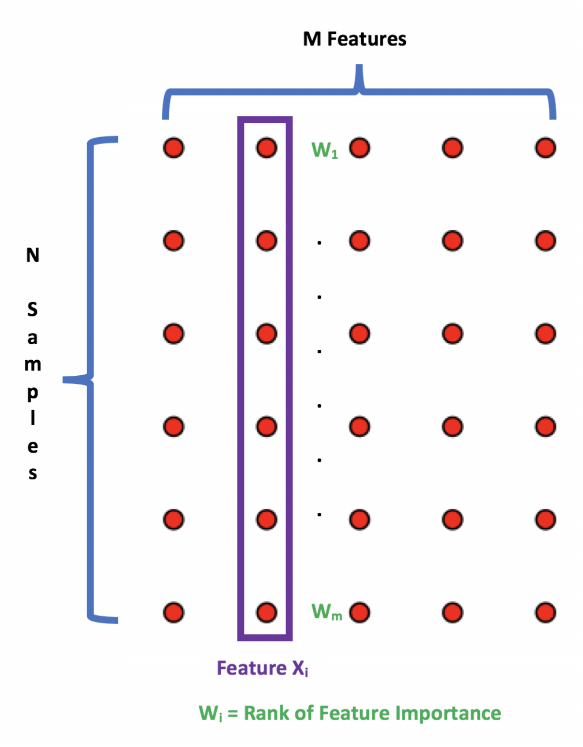

Figure 1: Rank by the feature importance

For rule-based decision trees, we directly employ the feature importance score for different features in random forest algorithm. This metric reflects the relative importance of all features involved in the classifier, and reveals connection between features X_i and Y.

After we have determined all feature importance values, we rank all the features by an increasing order of the absolute values of feature importance scores, because higher values indicate greater statistical significance in either positive or negative directions. Then, we assign to each feature X_i a weight w_i, equal to the rank by increasing values of the feature importance scores. The weight w_i reflects the significance of feature X_i in the classifier. Figure 1 demonstrates the process.

Once all w_i's have been constructed, we examine the correlation coefficients of feature X_i for each data point j, denoted by x_{ij}. If a feature X_i has positive correlation with the vector Y, we rank all data points by *decreasing* order of their corresponding x_{ij}'s, and define r_{ij} as the *rank* of x_{ij} in the set of all values of X_i's. Alternatively, if a feature has negative correlation, the the data is ranked in *increasing* order of x_{ij}, and r_{ij}'s are defined accordingly. Intuitively, r_{ij}'s show how much influence each feature X_i in data point j has to the final classification prediction. These ranks are constructed in a way to make sure that the data points with higher values of X_i are given greater consideration, since higher feature values in sociological datasets often correspond to special cases requiring extra attention.

Finally, for each attribute X_i in corresponding to data point j, we define the *feature importance index* with respect to X_i and data point j as

$$r'_{ij} = w_i r_{ij},$$

and denote $\mathbf{r}'_j$ as the *feature importance vector* consisting of r'_{ij}'s for data point j. The feature importance vector reveals information about the relative importance of data point j, and such information will be used to construct fairgroups for subsequent classification with fairness.

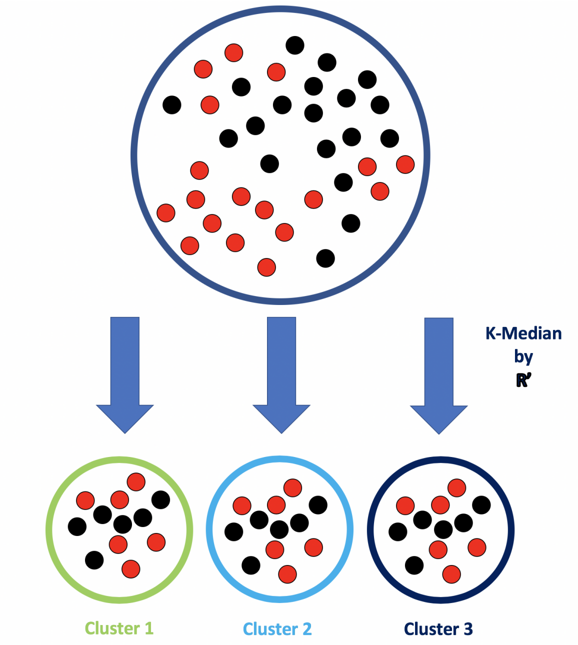

Figure 2: Median Clustering

Fairgroup Construction

With all feature importance vectors computed, we now examine how *close* these data points are with respect to these vectors, and how data points with similar features can be grouped together for easier analysis. To achieve these goals, we define a suitable distance between two vectors and consider a clustering problem where similar data points are grouped together.

Notice that each of the entries in the feature importance vectors are integers corresponding to different rankings, and that closer ranks imply similarity in one feature. Thus, we make use of the Manhattan-L1 distance to describe the distance between feature importance vectors $\mathbf{r}'_p, \mathbf{r}'_q$:

$$d(\mathbf{r}'_p, \mathbf{r}'_q) = \sum_{i=1}^{N} |r'_{ip} - r'_{iq}| = \sum_{i=1}^{N} w_i |r_{ip} - r_{iq}|,$$

Here N refers to the number of unprotected features.

Afterwards, we consider a K-median cluster algorithm to divide the entire dataset into K groups, each containing points with similar feature values as demonstrated by Figure 2. Notice that compared to other possible methods of clustering, K-median [Schmidt *et al.*, 2018] clustering is desirable for our purposes because it is robust to outliers and more adaptable to different types of distances.

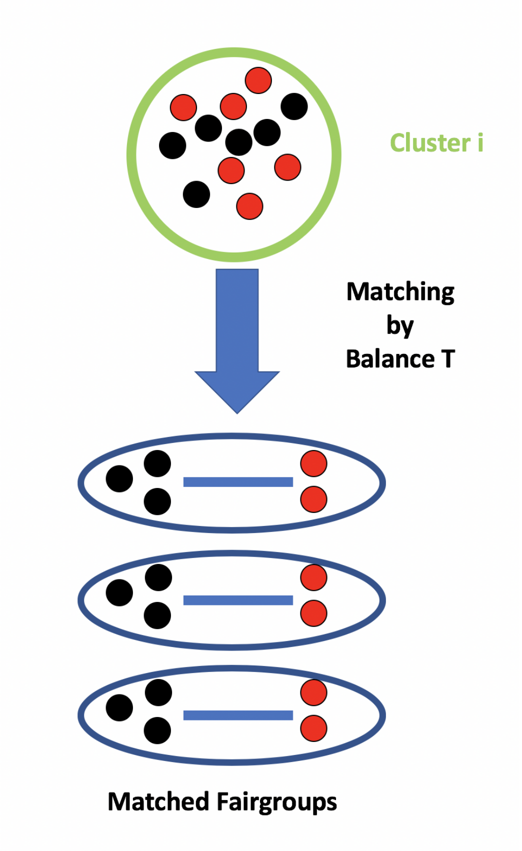

Figure 3: Group Matching

Within each cluster, we look at the protected feature. Without loss of generality, we assume that the protected feature is binary, and that our goal is to maintain that the balance of the protected feature A does not go below a predefined threshold t. Since this requirement implies that the ratio between $\#\{A = 0\}$ and $\#\{A = 1\}$ falls between t and $\frac{1}{t}$, we iteratively match as many $A = 0$ and $A = 1$ data points as possible, on condition that the ratio between $\#\{A = 0\}$ and $\#\{A = 1\}$ in each match falls between t and $1/t$. A set consisting of data points in such matches is denoted as a *fairgroup*. Figure 3 demonstrates the whole process.

To maintain as high accuracy as possible, we greedily construct fairgroups by first expressing t and $\frac{1}{t}$ as ratios $\frac{p}{q}$ and $\frac{q}{p}$, where p, q are co-prime positive integers. Starting from $\frac{\#\{A=0\}}{\#\{A=1\}}$, we iteratively match p data points where $A = 0$ with q data points where $A = 1$ (or q data points where $A = 0$ with p data points where $A = 1$) depending on whether $\frac{p}{q}$ or $\frac{q}{p}$ is smaller than and closer to the ratio of unmatched $\frac{\#\{A=0\}}{\#\{A=1\}}$. These matched $p + q$ points will form a fairgroup, and corresponding numbers of points will be moved from the unmatched point set. We repeat the procedure until all the points are matched or unmatchable. This procedure ensures that we create maximal numbers of fairgroups, so that even when one fairgroup is misclassified, the effects on the overall fairness and consistency are minimal.

Fairgroup-based Classification

For each fair-group we have thus constructed, we randomly pick a point from this fairgroup, and send this point to be classified by classifier C. Once a point is labeled, we need to take into consideration the properties of the protected feature to determine whether other data points in the same fair-group will be given the same label. For instance, in the case of SNAP distribution, protected features such as poverty should be treated as a protected feature only in the positive class, because the primary goal is to ensure that people receiving SNAP are mainly those living in poverty. On the other hand, for decision problems that ask for fairness in different label classes, the action should be extended to both positive and negative classes. While determining admission into selective schools, for instance, it is important that the odds of being admitted and rejected are roughly the same across different demographic groups to ensure equality. Figure 4 illustrates the process.

4 Experiments

4.1 Datasets and Preprocessing

For our experiments, we have focused on the United States Census American Community Survey data [Bureau, 2017], and considered two separate sub-datasets: the Medicaid dataset and the SNAP dataset. The Medicaid dataset consists of over 14,691,835 entries matching the individual level microdata. Each entry has 286 features, including an medicaid-receipt indicator. The SNAP dataset consists of 7,487,361 similar entries. Each entry contains 286 variables, including a SNAP-receipt indicator.

Features and Data Cleaning

It is important to pick out the importance features to build the model precisely. Although the dataset itself has 286 features in total, only a portion is relevant. For instance, in the case of Medicaid, features such as *if your family owns an air-conditioner* or *the number of bedrooms* are not related to the final approval decision, and are be filtered out.

Protected Variables

Experiments suggest that the feature household income and the poverty level are of the highest importance for both Medicaid data and SNAP data. Other variables include disability, number of persons in a household, poverty status, locations, etc, shows less importance for the decision of Medicaid. Preliminary experiments suggest that income and poverty level are most closely related to the target variable. Thus, in the following experiments, we will use *income* or *poverty* as protected variables for both of the datasets.

Target Variable

The target variable is the output decision variable of the given model. In our experiments, the target variables for the classifier are the numerical vectors indicating medicaid/SNAP receipt.

4.2 Experimental Results

We have tested on two types of the most popular classifiers: decision trees and regression models. For regression models

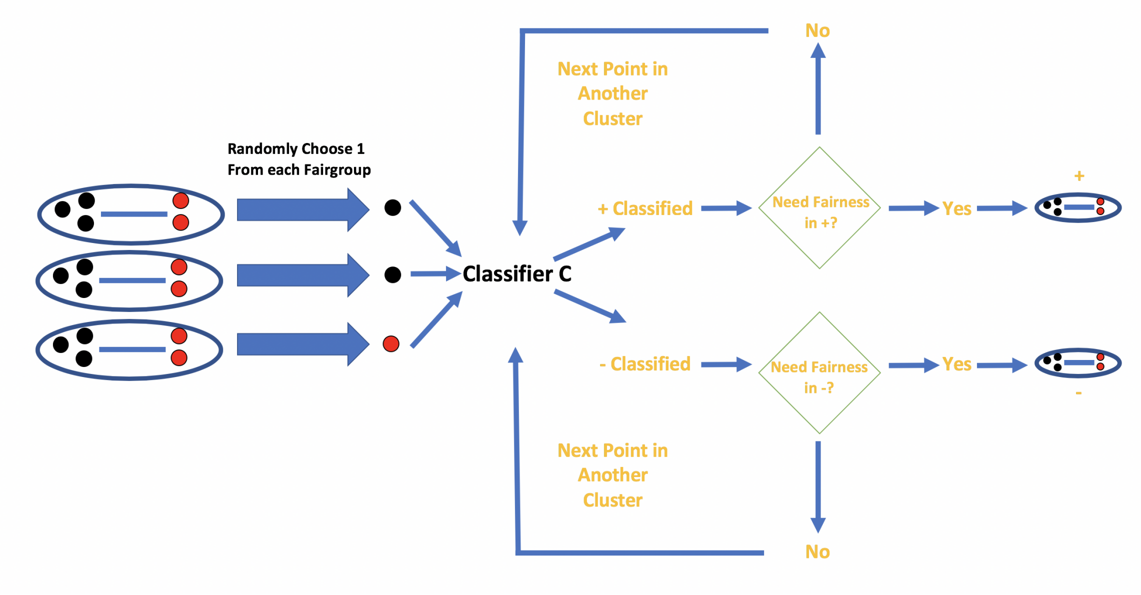

Figure 4: Classification with respect to each fairgroup

we consider linear regression, logistic regression and support vector machine(SVM). For decision trees we consider random forests. In total we have carried out 2 sets of experiments, each involving 4 types of classifiers.

For both of the datasets, we have split our data into training and testing sets. We first apply our algorithm to the training dataset to obtain feature importance scores corresponding to each feature. Once we have selected the protected feature (feature with largest importance score) of income, we group the entire dataset into 5 clusters by K-median clustering [Zhu and Shi, 2015]. The choice of K is determined by the standard choice of cluster numbers yielding the best empirical evaluations. In each cluster, we maintain the same ratio for poverty and non-poverty households by setting the balance as $\frac{2}{8} = \frac{1}{4}$ between poverty and non-poverty households, and iteratively match points accordingly.

In the Medicaid dataset, our matching algorithms produce a match such that more than 80 percent of recipients live in poverty. In contrast, standard classifiers without our classification algorithm produce a outcome such that only less than 70 percent of recipients are actually in poverty. Compared to the case without fairgroup construction, our method demonstrates greater fairness and allocates resource more properly by ensuring that the majority of households receiving medicaid are indeed in poverty. Meanwhile, the classification accuracy of the classifiers with our processing algorithm is still comparable without our algorithm. Table 1 and Table 2 list the detailed numerical results of our experiments. Same experiments have been carried out for the SNAP data. Our algorithm produces matchings such that around 80 percent of SNAP receipts live in poverty. In contrast, standard classifiers yield matchings where around 40 percent of SNAP receipts live in poverty. Table 3 and Table 4 list our results in more details. Figures 5 to 8 visualize the increase of fairness in classification.

Model	Poverty Rate	Accuracy
Random Forest	68.3	93.1
Logistic	67.4	92.6
LR	65.3	90.2
SVM	68.7	91.5
RF + Fairgroup	**85.7**	90.1
Logistic + Fairgroup	84.3	89.5
LR + Fairgroup	82.7	88.1
SVM + Fairgroup	83.1	88.3

Table 1: Experimental results on Medicaid with Protected feature of lower income

Model	Poverty Rate	Accuracy
Random Forest	68.3	93.1
Logistic	67.4	92.6
Linear Regression	65.3	90.2
SVM	68.7	91.5
RF + Fairgroup	**87.5**	90.1
Logistic + Fairgroup	84.7	89.3
LR + Fairgroup	83.4	86.9
SVM + Fairgroup	83.6	88.9

Table 2: Experimental results on Medicaid with Protected feature of Poverty Level

4.3 Result Analysis

The results demonstrate significant increases on the fairness when our algorithm is applied. These increases mean that more people will be allocated with the resources the need by true necessity. The results from these tables also suggest that Random Forests, in combination with our algorithm, achieve the best result with a reasonable tradeoff for the accuracy.

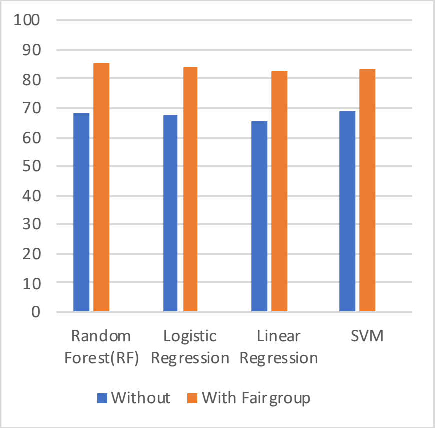

Figure 5: Medicaid - Income

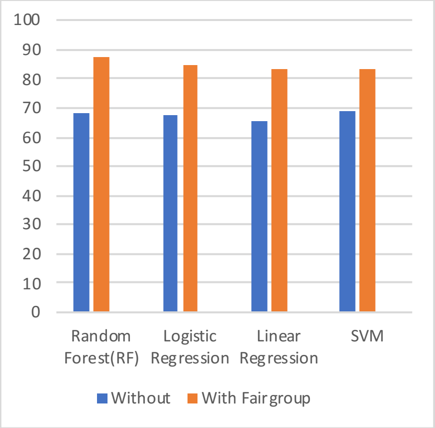

Figure 6: Medicaid - Poverty

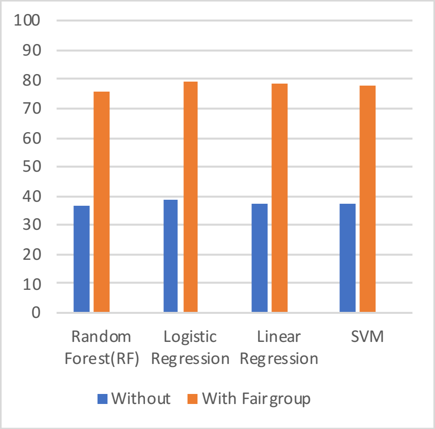

Figure 7: SNAP - Income

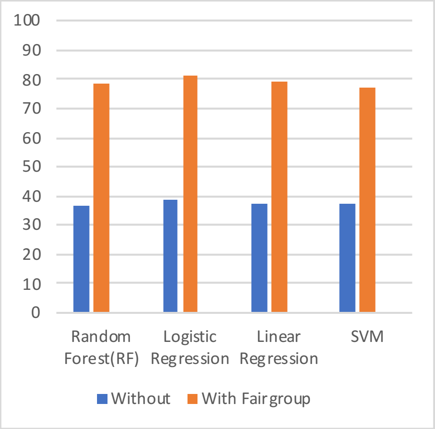

Figure 8: SNAP - Poverty

Model	Poverty Rate	Accuracy
Random Forest	36.4	93.1
Logistic	38.8	92.6
Linear Regression	37.1	91.7
SVM	37.6	91.4
RF + Fairgroup	75.7	87.2
LOGISTIC + Fairgroup	**79.3**	88.5
LR + Fairgroup	78.4	85.9
SVM + Fairgroup	77.9	87.9

Table 3: Experimental results on SNAP with Protected feature of lower income

Furthermore, comparing model accuracies with/without our algorithm, we can see from that our fairness prediction model does not hurt the accuracy of model, thereby boosting the validity of our algorithm.

5 Conclusions

In this work we present a novel algorithm to improve fairness of machine learning classifiers for social decision problems. To achieve our goal, we propose *fairgroup* construction to emphasize protected variables in the classification process. Experiments on real datasets demonstrate our algorithm's effectiveness in boosting fairness, while maintaining high classification accuracies.

Model	Poverty Rate	Accuracy
Random Forest	36.4	93.1
Logistic	38.8	92.6
Linear Regression	37.1	91.7
SVM	37.6	91.4
RF + Fairgroup	78.4	87.2
LOGISTIC + Fairgroup	**81.4**	88.5
LR + Fairgroup	79.1	86.7
SVM + Fairgroup	77.2	88.1

Table 4: Experimental results on SNAP with Protected feature of Poverty Level

Acknowledgments

We thank Prof Yi Fang at Santa Clara University for proofreading our work. Boli Fang is supported in part by NSF CCF-1844234 and IIS-1633215.

References

[Bureau, 2017] US Census Bureau. American community survey 2017 5-year estimate. 2017.

[Chierichetti *et al.*, 2017] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[Corbett-Davies and Goel, 2018] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

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

[Joseph *et al.*, 2016] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 325–333. Curran Associates, Inc., 2016.

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

[Kleinberg *et al.*, 2016] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[Morse, 2018] Susan Morse. Artificial intelligence helps insurers identify medicare members who also qualify for medicaid, Nov 2018.

[Nations, 2016] United Nations. 17 goals to transform the world for persons with disabilities. 2016.

[Rösner and Schmidt, 2018] Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. 2018.

[Schmidt *et al.*, 2018] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means clustering. 2018.

[Zadrozny, 2004] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 114–, New York, NY, USA, 2004. ACM.

[Zafar *et al.*, 2015] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2015.

[Zhu and Shi, 2015] Haoyu Zhu and Yuhui Shi. Brain storm optimization algorithms with k-medians clustering algorithms. In *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, pages 107–110. IEEE, 2015.

[Zink and Rose, 2019] Anna Zink and Sherri Rose. Fair regression for health care spending. *arXiv preprint arXiv:1901.10566*, 2019.