Achieving Fairness in Determining Medicaid Eligibility through Fairgroup Construction

Boli Fang[1] , **Miao Jiang**[1] , **Jerry Shen**[2] , **Pei-Yi Cheng**[1] and **Manju Chivukula**[1]
[1]Department of Computer Science, Indiana University, USA
[2]Price School of Public Policy, University of Southern California, USA
{bfang, miajiang, peicheng, mchivuku}@iu.edu, haoxuans@usc.edu

Abstract

Effective complements to human judgment, artificial intelligence techniques have started to aid human decisions in complicated social problems across the world. In the context of United States for instance, automated ML/DL classfication models, through quantitative modeling, have the potential to improve upon human decisions in determining Medicaid eligibility. 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 the fairness in classification outcomes. Experiments on American Community Survey dataset demonstrate that our method effectively boosts the fairness of decision making in determining Medicaid eligibility, while maintaining very high accuracy comparable with that achieved by original classifiers.

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. In particular, proper allocation of health and medical resources are vital for the wellbeing of citizens across different countries. While the majority of previous endeavors have centered on the developing world, one cannot ignore the related issues in developed countries. According to the American Community Survey [Bureau, 2017], millions of American households are regularly receiving governmental assistance in receiving Medicaid, a compensation scheme designated for low-income individuals to receive proper reimbursement for necessary medical treatment. It is noted in the same dataset that over 16 million households in America are living ”below poverty level”, yet a substantial amount of poor households are not yet receiving Medicaid. 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]. Such disparity and inequality behoove decision makers to introduce complementary policies that better take various factors involved in Medicaid Eligibility 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. Taking into consideration aspects of computational actions and socioeconomic context, previous researchers have focused on two subcategories of fairness as benchmarks - outcome fairness and process fairness. Given the nature of most social welfare programs, which lean towards benefiting individuals and households with certain(often disadvantaged) socioeconomic statuses, outcome fairness is often more important than process fairness under such scenarios. Depending on the nature of the problem, one can group the factors into two categories: *protected* factors which is of priority in determining fairness and *unprotected* factors which doesn't carry as much priority. 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 fairly distribute Medicaid resources among individuals and households, while maintaining a high degree of classification 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 outcome classes. Our contributions in this work can be summarized as follows:

1. We provide an outcome-fairness algorithm for the allocation of Medicaid resources by defining fairgroups, and achieves fairness with respect to the protected features, in the Medicaid Decision Problem.

2. Our algorithm also takes into consideration unprotected features while making decisions on fairness, so that individuals with similar features will be still classified in similar ways and the overall classification accuracy remains high.
3. The method to achieve fairness as introduced in our paper is easily adaptable to other decision making problems involving the distribution of scarce resources, such as Judicial Decisions, acceptance to educational programs and approval of credit card.

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 [Chierichetti *et al.*, 2017] have been produced. [Kleinberg *et al.*, 2016] suggested that although it's not possible to achieve some desired properties of fairness at the same time, including "protected" features in algorithms would increase the equity and efficiency of models. Grgic-Hlaca et. al. (2016) previously discussed three methods of measuring process fairness - feature-apriori fairness, feature-accuracy fairness, and feature-disparity fairness.

The second group has centered on algorithms to achieve fairness. Along the route of disparate impact, [Feldman *et al.*, 2015] has described algorithms to spot the presence of disparate impact through Support Vector Machine, while [Chierichetti *et al.*, 2017] applied the notion of disparate impact to design an algorithm that achieves balance in unsupervised clustering algorithms. This paper also introduces the notion of *protected and unprotected features* which will used in our paper.

3 Model

In this section we present a novel strategy by constructing *fair-groups* to achieve fairness in classification results. 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 Preliminaries

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, without loss of generality and for sake of simplicity, that the protected traits are binary and that the classification label class is also 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 $\#\{A = 0\} = \#\{A = 1\}$. A dataset is 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. Following the doctrine of disparate impact as stated in [Chierichetti *et al.*, 2017], we 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.

3.2 Fair-group construction

We provide in this section the details of the algorithms we will use to achieve fairness in classification. Assume that we already have a classifier C which yields predictions for data points and might not yield α-fair classification results. Overall, our algorithm constructs fair-groups from testing data, and conducts classification on the data points with C while taking the properties of the fairgroups into consideration.

The sections below provide more details of our method.

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. A natural choice is the feature importance score [Hastie *et al.*, 2009] of features X_i in the classification model, because each score determines the contribution of each feature to the final classification outcome.

We then rank all the features by an increasing order of the absolute values of feature impotance scores coefficients, because higher correlation values indicate greater statistical significance in either positive or negative directions. Then, we assign to each feature X_i a weight w_i which is 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.

After constructing the relative weight w_i of each feature X_i, we examine the actual values of X_i for each data point j, here denoted by x_{ij}. If a feature X_i has positive correlation with Y, then we rank all data by the decreasing order of the corresponding x_{ij}'s of the feature X_i, 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, the rank 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 enough consideration, since higher feature values in socialogical datasets are often likely to correspond to special cases requiring extra attention. Finally, for each attribute X_i in corresponding to data point j, we define $r'_{ij} = w_i r_{ij}$ as the *feature importance index*, and define $\mathbf{r}'_j$ as the *feature importance vector* corresponding to 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 fair classification.

Fairgroup construction

With each data point now represented in the form of feature importance vectors, we now examine how close these data points are in terms of the influence each data point might exert to the final classification outcome, 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. Within each cluster, we look at the protected features. Without loss of generality, we assume that the protected feature is binary, and that our goal is to maintain the balance of the protected feature A does not go below a certain threshold t. Since this requirement implies that the ratio between $\#\{A = 0\}$ and $\#\{A = 1\}$ falls between t and $\frac{1}{t}$, we 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*.

Classification with respect to each fairgroup

For each fair-group we have thus constructed, we randomly pick a point to be classified by C. If the point is labeled as $+$, we apply the same label to all other data points in the group. Alternatively, if the point is labeled as $-$, 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 Food Stamp distribution, protected features such as poverty should be treated as a protected feature only in the positive label class, because our primary goal is to ensure that people receiving food stamps are mainly composed of people living under the poverty threshold. On the other hand, for decision problems that favor similar representation of one feature in different label classes, we need to include the feature in both positive and negative classes. While determining admission eligibility for 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.

Moreover, to reduce the negative effect of potential misclassification as much as possible, we construct as many fairgroups as possible by first expressing t and $\frac{1}{t}$ as ratios $\frac{p}{q}$ and $\frac{q}{p}$, where p, q are co-prime 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 $A = 0$, $A = 1$ 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 due to the misclassification of the randomly drawn point, the effects on the overall fairness and consistency can be minimal.

4 Experiments

4.1 Dataset

To conduct experiments using the model explained above, we focus on the United States Census American Community Survey data [Bureau, 2017]. Consisting over 7487361 entries, the individual level microdata displays various potentially useful features, including status of receiving medicaid. For each entry, there are 286 variables, including an indicator if the medicaid is given or not.

Features and data cleaning

Before implementing random forest onto the data set, we need to pick out the importance features to build the model precisely. Although the dataset itself has 286 features in total, only a portion of them are related to the decision of issuing Medicaid, and thus data cleaning is required. We first exclude some features by common sense, such as *if your family owns an air-conditioner* or *the number of bedrooms*, which are obviously not related to the decision of Medicaid. Table 1 provides a list of important features we have ever used, and an example of real value for each of the example feature.

Feature	Example
Age	25
Division	1-New England
Region	2-Midwest
State	26-Michigan
Gender	2-Female
Number of Children	2
Hearing Difficulty	1-Yes
Vision Difficulty	2-No
Ambulatory difficulty	1-Yes
Self-care difficulty	2-No
Class of workers	4-State Employee
Household Income	$25,000
Interest Income	$5,000
Race	3 - American Indian alone
Marital Status	1-married
Poverty Status	1-Yes

Table 1: Features Used in the Experiment

After the initial feature filtering, we need to decide which feature to be the protected variable. As we mentioned, the protected variable should be one that should be of most relevance to the medicaid decision. Here for the most effective prediction, we have applied random forest with the selected features on the entire dataset, and computed the feature importance scores accordingly. Experiments suggest that the feature household income is of the highest importance. Other variables include disability, number of persons in a household, poverty status, locations, etc, shows less importance for the decision of Medicaid. Table 2 lists the importance of some of the features that we are going to use in our analysis. In the following experiments, we will use the income as the protected variable, although similar methods can also apply to other features of interests as well.

Feature	Feature Importance
Age	0.0783
Division	0.00532
Region	0.00132
State	0.00197
Gender	0.00215
Number of Children	0.00306
Hearing Difficulty	0.0121
Vision Difficulty	0.0121
Ambulatory difficulty	0.0121
Self-care difficulty	0.0121
Class of workers	0.127
Household Income	**0.398**
Interest Income	0.211
Race	0.00587
Marital Status	0.0445
Poverty Status	0.0747

Table 2: Feature importance of some variables

Target Variable

Here in our experiment, the target variable for the classifier is the feature vector which indicates whether a single individual has finally received medicaid or not. Notice that the variable is binary, so decision tree and other classification algorithms follow naturally in our modeling.

4.2 Results

We have conducted two comparable experiments to show that our particular algorithm indeed does improve the fairness of the results, compared with the cases where classifiers such as logistic regression and Random Forest only are used.

Splitting our data into training and testing sets, we applied random forest 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 follow the algorithm described in the previous sections, and group the entire dataset into 5 clusters by K-median clustering [Zhu and Shi, 2015] as by the standard choice of cluster numbers in clustering algorithms. In each cluster, we maintain the same ratio for poverty and non-poverty households by setting the balance as $\frac{8}{2} = \frac{4}{1}$ between poverty and non-poverty households and matching points accordingly.

Under such settings, our experiments show that out of the people receiving Medicaid, 83.4 percent are living under poverty line. In contrast, standard classifiers without our classification algorithm produce a outcome such that out of the people who are receiving Medicaid, only less than 70 percent of households 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 after our processing algorithm is still comparable without our algorithm.

Method	% of Poverty	Model Accuracy
Pure Random Forest(RF)	68.3	93.1
RF + Fairgroup	**85.7**	90.1

Table 3: Experimental results

5 Conclusion

In this work we present a novel approach to solve the problem of Medicaid Eligibility Determination by introducing an outcome-fair algorithm over classifiers. To achieve our goal, we propose the strategy of *fair-group* construction, to promote representation of households in poverty in the group of people receiving Medicaid. Experiments on the US Census individual level microdata yields results that are more consistent among samples with similar attributes. As a part of our future work, we hope to apply our method to a wider range of classifiers, and address the current social problems related to inequality and inequity in both the developed and developing world.

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.

[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.

[Grgic-Hlaca *et al.*, 2016] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS Symposium on Machine Learning and the Law*, volume 1, page 2, 2016.

[Hastie *et al.*, 2009] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction, springer series in statistics, 2009.

[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.

[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.