Optimized Pre-Processing for Discrimination Prevention

Flavio P. Calmon
Harvard University
flavio@seas.harvard.edu

Dennis Wei
IBM Research AI
dwei@us.ibm.com

Bhanukiran Vinzamuri
IBM Research AI
bhanu.vinzamuri@ibm.com

Karthikeyan Natesan Ramamurthy
IBM Research AI
knatesa@us.ibm.com

Kush R. Varshney
IBM Research AI
krvarshn@us.ibm.com

1 Objective

This document presents the findings obtained using a data pre-processing algorithm for discrimination prevention we have recently developed(1). We also discuss the salient aspects of its software implementation [1].

2 Summary of the Proposed Algorithm and Our Findings

Our optimization method transforms the probability distribution of the input dataset into an output probability distribution subject to three objectives and constraints: (i) group discrimination control, (ii) individual distortion control, and (iii) utility preservation.

By group discrimination control, we mean that, on the average, a person will have a similar chance at receiving a favorable decision irrespective of membership in the protected or unprotected group. By individual distortion control, we mean that every combination of features undergoes only a small change during the transformation. Finally, by utility preservation, we mean that the input probability distribution and output probability distribution are statistically similar so that the AI algorithm can still learn what it is supposed to learn.

We take a very general and flexible optimization approach for achieving these objectives and constraints. All three objectives are mathematically encoded with the user's choice of distances or divergences between the appropriate probability distributions or samples. Our method is more general than previous work on pre-processing approaches for controlling discrimination, includes individual distortion control, and can handle multiple protected attributes.

We applied our method to two datasets: the ProPublica COMPAS prison recidivism dataset (an example containing a large amount of racial discrimination whose response variable is criminal re-offense) and the UCI Adult dataset based on the United States Census (a common dataset used by machine learning practitioners for testing purposes whose response variable is income). With both datasets, we are able to largely reduce the group discrimination without major reduction in the accuracy of classifiers such as logistic regression and random forests trained on the transformed data.

On the ProPublica dataset with race and gender as protected attributes, the transformation tends to reduce the recidivism rate for young African-American males more than any other group. On the Adult dataset, the transformation tends to increase the number of classifications as high income for two groups: well-educated older women and younger women with eight years of education.

[1] https://github.com/fair-preprocessing/NIPS2017

3 Code Description

We now describe the major components of our software implementation. The software has the following modules:

- **Data preparation:** We convert any numeric attributes in the dataset to corresponding nominal attributes. The end-user can use any standard method of making the data nominal as long as the number of categories within each attribute does not end up being pretty high. This is performed to simplify the subsequent steps of our algorithm.
- **Setting up the distortion metric:** This distortion function receives two sets of inputs - the original features before pre-processing and the features after pre-processing. This function penalizes undesirable distortions, by assigning costs to each of the possible feature value perturbations. The practitioner can set these costs based on the application.
- **Setting up the discrimination metric**: The practitioner can set the discrimination control parameter and the algorithm will initially determine if the solution lies in the feasible or infeasible region before proceeding to the subsequent steps.
- **Randomization step:** We split the original dataset into train and test sets and apply our randomized mapping to obtain the pre-processed train and test sets, respectively. The randomized mapping is learned by solving the optimization problem described earlier.
- **Applying Classifiers:** Our prediction model is built using the pre-processed training set and this is applied for predicting the classes on the pre-processed test set. A key feature differentiating the training and testing sets is that the former has class labels that will also be transformed, whereas the latter does not.

References

[1] Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K.N. and Varshney, K.R., 2017. Optimized Pre-Processing for Discrimination Prevention. In *Advances in Neural Information Processing Systems* (pp. 3995-4004).