Obtaining Fairness in Machine Learning using Optimal Transport Theory

Prof. J.Saketha Nath,
Computer Science and Engineering,
Indian Institute of Technology, Hyderabad

Prof. Ganesh Ghalme,
Artificial Intelligence,
Indian Institute of Technology, Hyderabad

R. Gokul Kannan,
4th year B.Tech in Computer Science and Engineering,
Indian Institute of Technology, Hyderabad

Paper followed can be found here.
The code that I wrote for this project can be found in this GitHub repository.

Are machine learning models fair? This has been a problem for many years (since 2012-13).

When a machine learns from the given training data using some machine learning algorithm, there may be some bias in that data, which is also learnt by it. This causes unfairness for future data we may be testing on.

People thought of different ways to unbias the data.

1) Individual fairness (Dwork et al) - Similar individuals are treated similarly
2) Group Fairness (Gender, Race, etc.)

Group Fairness:

One way is to ignore the feature which causes the bias, but that is not a viable option, as other features may indirectly depend on the mentioned feature. If we choose to ignore all the dependent features too, we may be left with no/very less train data and thus can't train properly. (Maybe in the past i.e. train data, only one group was considered suitable for a job, but now that is not the case).

Some other ways to try to achieve fairness are as follows:

- Change the algorithm itself, at the cost of its accuracy.
- Change the data itself.
- Flip some outputs of the algorithm to maintain fairness.

These are the in-processing, pre-processing, post-processing methods.

a) Outcome based fairness

b) Treatment based fairness - explainable AI

Fairness Notion in Group Fairness:

1) Accuracy equality - This notion tells that across different groups the performance of an algorithm should be equal.

2) Statistical Parity - Fraction of positive classification should be equal across the groups

3) Disparity (Extension of 2) - Minimum of ratio of positive classification across the 2 groups should be greater than a constant (typically 0.8), else it is unfair.

$$DI(g, X, S) = \frac{\mathbb{P}(g(X) = 1 \mid S = 0)}{\mathbb{P}(g(X) = 1 \mid S = 1)}.$$

This formula is directly a measure of fairness of our classifier. It is known as Disparate Impact of the classifier g with respect to the data (X, S). When statistical parity is achieved, DI will become equal to 1, but it is often unrealistic and we will relax it into achieving a certain level of fairness. (We take 0.8 which is also known in the literature as the 80% rule).

4) Balanced Error rates - The mistakes we make across different groups have to be balanced. (The algorithm's misclassification rate is not different based on the group).

$$BER(g, X, S) = \frac{\mathbb{P}\left(g(X) = 0 \mid S = 1\right) + \mathbb{P}\left(g(X) = 1 \mid S = 0\right)}{2}.$$

Here, g is the Classifier we are using, X is the attributes, and S is the protected attribute.

Optimal Transport Theory:

Study of transportation and allocation of resources optimally. Classic example is moving iron ores out of m mines to n factories, in such a way that the total cost taken to move them is minimum. We will be using it to modify the data in our problem.

OT has various applications like: Changing colour range of images (using colours from one image to repaint another image, there exists a cost function for changing one colour to another), Domain Adaptation in Machine learning (Each point from one domain is mapped to another point in second domain, distance between them can be taken as the cost function), Shape Interpolation, Distributional robustness, etc.

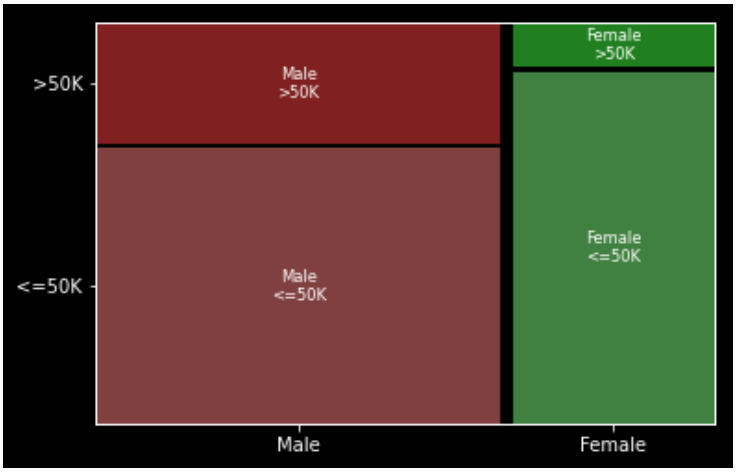
Removing Disparate impact using Optimal Transport:

We will be using the adult dataset for analysing fairness in this project. It contains about 48k records. We will train a classifier and use it to predict if the given person will have an annual income of above 50k dollars or not. We have chosen the columns age, educational-num, gender, capital-gain, capital-loss, hours-per-week, income according to the paper.

Here, the notion of fairness is “will our model predict that a person gets less than 50k annual income per year, only because that person is a woman?”.

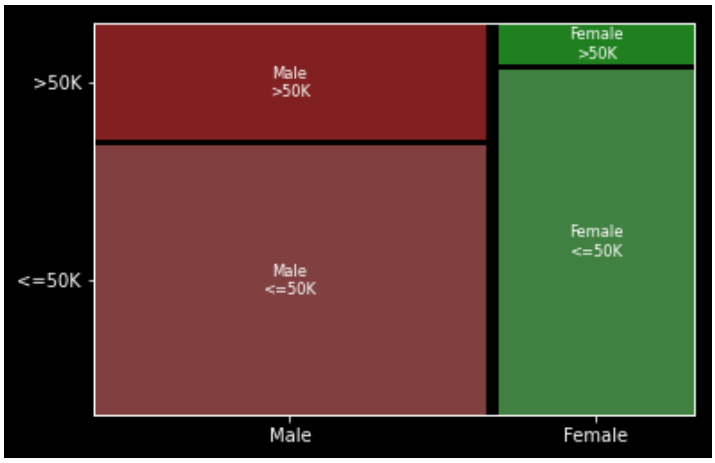
Now we have 2 distributions of data, the men and women. If we find a way to change the women’s attributes to that of men’s in such a way that they are as close as possible, we can use that modified data to train our classifier. First we tried using the Wasserstein barycenter, to get the common barycenter of those distributions. But then we realised, we will not have a way to get the actual transport plan. This is a classic Optimal Transport problem. So we have used the python OT library for this.

As the dataset is large (48k) and computing cost matrix and doing OT is hard for the whole dataset, we have sampled 10000 records from the dataset and used it in the project. The mosaic plot of the dataset is as shown:



We can see that the fraction of men with annual salary >=50k is high compared to the fraction of the same for women. So due to this bias, our model will not be fair. We make sure that the 5k samples we take also have a similar fraction of men and women, and use it for the project.

Out of that 10,000 samples there were 6674 men and 3326 women. We get the following mosaic plot for the samples I took:

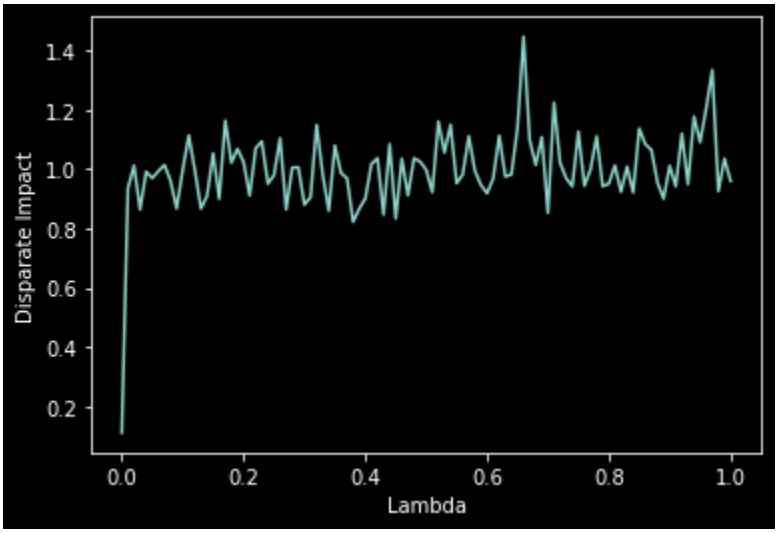


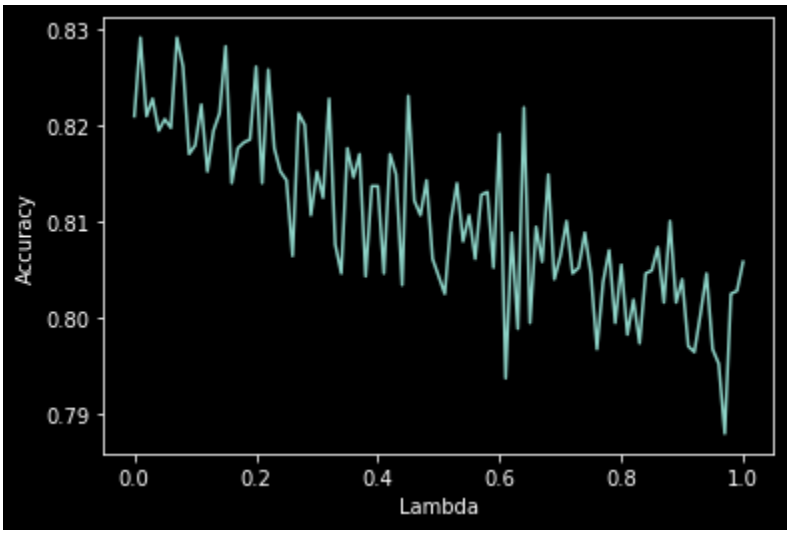
Did one hot encoding of men as 1, women as 0, >=50k as 1 and <50k as 0.

We take a matrix of women along the rows and men along the columns, compute the cost matrix and find with which probability each woman corresponds to each man using OT.

We have used a Logistic Regression Classifier to train and test (with solver as lbfgs and max_iter as 10000). We have also done scaling of the features so one feature will not contribute more to the computation of the cost matrix. We have used test-train split from sklearn to split the data into train and test sets with a fraction of 0.33.

We define a parameter λ according to which we will decide whether to replace a woman's record with the equivalent man's record. We replace it with probability λ. We vary this value of lambda and see how it affects the accuracy and disparate impact. I tested for 100 values of λ to see how DI and accuracies vary with it. The following are the graphs that we got:





If λ=0, it means we aren't replacing any data i.e. we are using the original data itself. DI was 0.11073189018010508 and accuracy was 82.1% approximately.

Ideally, the accuracy should be monotonically decreasing and DI should be monotonically increasing, but due to various factors (like sampling, replacing, test-train split, etc.) there are anomalies.

I tried many things, like replacing in place, retraining the classifier, scaling at different times, but I could not find why there was a sudden jump in DI when there is not much change in Lambda.

Please refer to the actual whole file mentioned in the start to see the actual values of DI, accuracy for each lambda. As there were 100 values of each, I couldn't tabulate them here.

Future Improvements:

If there's more computation power available, the whole data can be used for the model, instead of sampling.

One idea I had was to split into train, test sets and perform OT on them, repair them separately and use the same train set to train, and same test set to test, before and after OT. That way the leaking of data from test set to train (while scaling and while repairing data) can be avoided.

Maybe we can try including more features. But in the paper we followed, they took only these 6 features.