Université de Lille

Master 1 - Data Science

Research Project

Ensuring Metric-optimal Fairness for Online Decision Making

based on

Metric-Free Individual Fairness in Online Learning, Bechavod et al. [1]
&
A Statistical Test for Probabilistic Fairness , Taskesen et al. [2]

Author:
Maxime BOUTON

Supervisor:
Debabrota BASU

April 8, 2021

Abstract

This research project aims to verify and use the statements of two theoretical papers about fairness in decision making. The first one is mostly about individual fairness while the second one deals with group fairness.

I want to thank the authors of [2] for letting me get an access to their code, as well as my supervisor Debrabrota BASU for helping me out through this research project, initially designed for the second year students of this Master Data Science.

Contents

List of Figures **ii**

List of Algorithms **ii**

1 Introduction **1**

1.1 Bias and heterogeneity in the data 1

1.2 Fairness in ML 2

2 Individual fairness study **3**

2.1 Definitions 3

2.2 Fair classification 4

2.3 Implementation of the reduction 6

2.4 Numerical results and conclusion 6

2.4.1 Offline classification 6

2.4.2 Online classification 6

2.4.3 Reduction from online classification 6

3 Group fairness study **7**

3.1 Definitions 7

3.2 Optimal Transport (OT) to test group fairness 8

3.2.1 A quick presentation of OT 8

3.2.2 Wasserstein distance 8

3.2.3 Principle for testing 9

3.3 The general idea and the actual implementation 10

3.4 Results 10

4 General conclusion, improvements and personal comments **11**

References **15**

List of Figures

1 Illustration of bias cycle in ML process from [3] 1
2 Illustration of Simpson's Paradox from [3] 2
3 Dataset presentation 6
4 Online classifier score 7
6 Illustration of Optimal Transport 9
5 Reduction score 12
7 Score comparison between fair and training classifier 13
8 Fairness comparison between fair and training classifier 13
9 Dataset pair-plots 14

List of Algorithms

1 Reduction from Online Fair Batch Classification to Online Batch Classification, from [1] 5

1 Introduction

Through the years, Machine Learning (ML) and more specifically automated systems for decision making have grown more and more popular.

Such systems can indeed be found everywhere, in any fields of society, from education to healthcare, through criminal justice, lending... That is why, their decisions can have life changing consequences regarding their purpose. Therefore, one cannot blindly rely on those, unless they were proven efficient enough, unbiased and above all fair.

This article aims to create a code-base of online classification and statistical measures of fairness, and test it on real data sets relating to real applications.

1.1 Bias and heterogeneity in the data

As detailed in [3], the data used for training of AI systems are most of the time full of bias of any kind. Unfortunately, the algorithms that proceed the data, are more likely to reproduce the imbalance it contains.

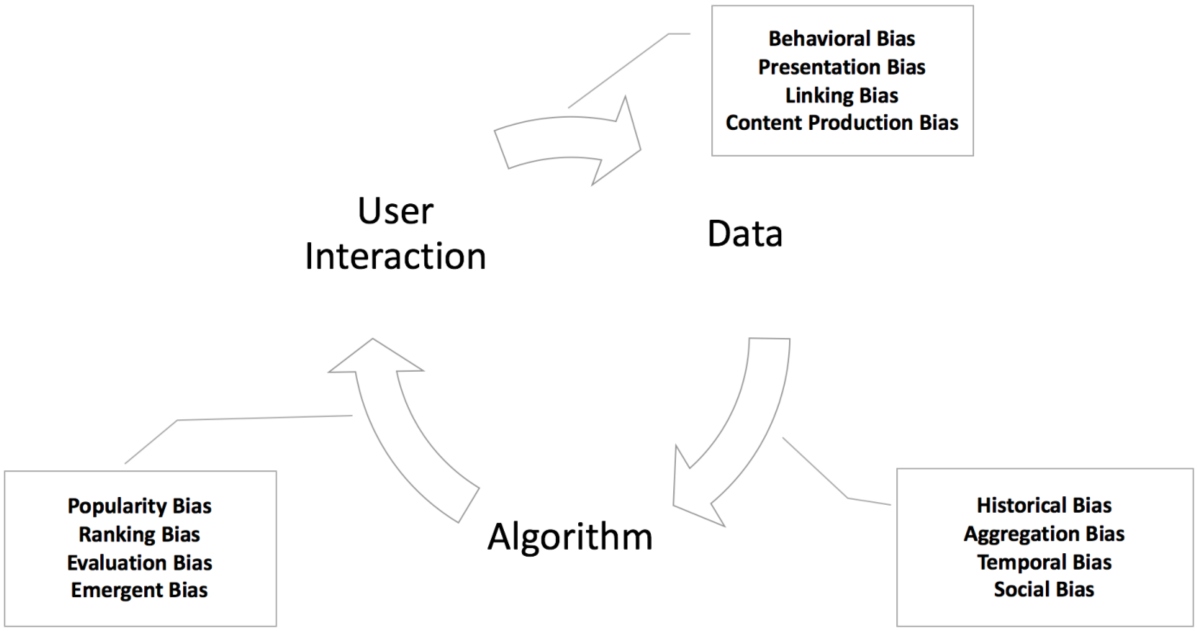


Figure 1: Illustration of bias cycle in ML process from [3]

The first difficulty is not about avoiding bias in the data, but it is to know whether there is actually bias. The Simpson's paradox is a good illustration of such underlying bias, if one is not aware of how the data and sampling was actually done.

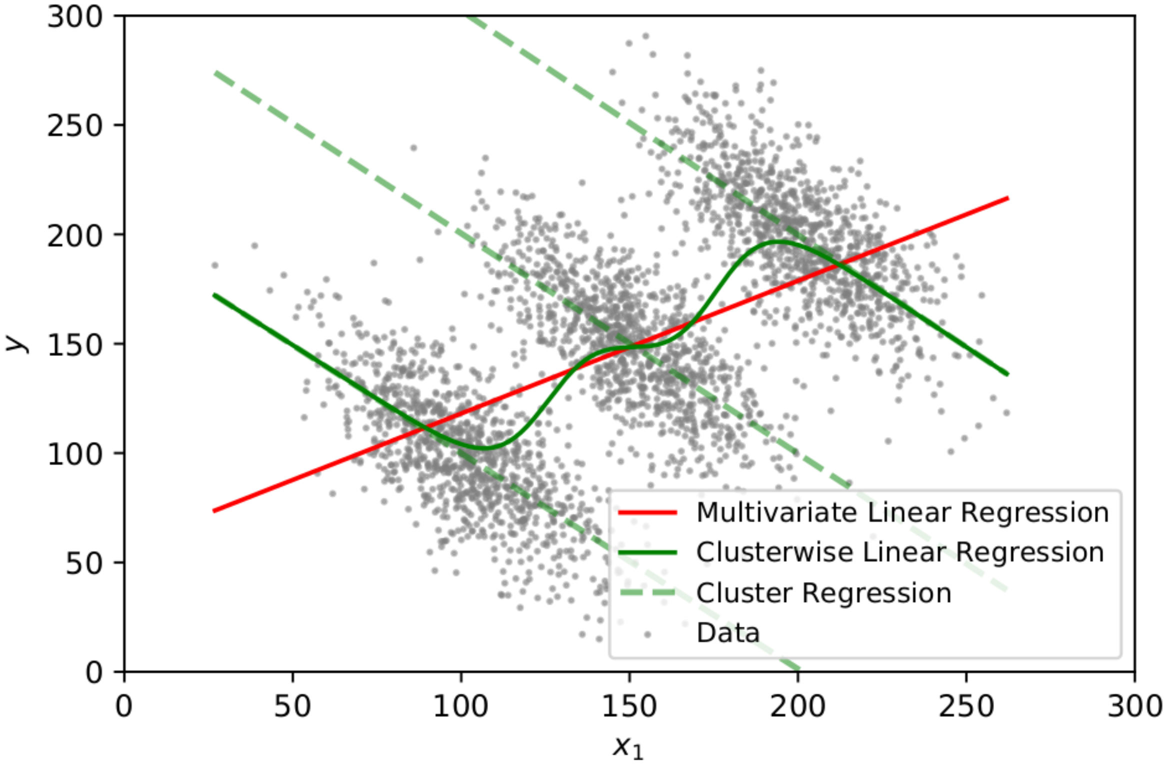


Figure 2: Illustration of Simpson's Paradox from [3]

This study was designed to prove that the average tendency of the entire group is not necessarily the one of the different subgroups. Here, the dataset is taken from a nutrition study : x represents the daily pasta consumption and y is the body mass index (BMI).

The red curve, which is regression over the whole population, tends to prove that the higher the pasta consumption, the higher the BMI. When looking closer at the population subgroups, that are related to one's level of sport activity, the tendency is the opposite.

This is therefore a good approach to understand why it is important to know the underlying distribution of the population.

1.2 Fairness in ML

Fairness is one of the most important features in ML, because when it comes to real life decisions, the result displayed by the artificial intelligence (AI) must not reflect any discriminatory behaviour present in its training data.

As seen above, there are risks when the distribution of the data is not perfectly known, which is of course the case most of the time for real world applications. Those underlying factors can therefore lead to unfair decisions with respect to different individuals (*individual fairness*), or different subgroups among the whole population (*group fairness*).

Fairness is a very complex notion, that is really hard to translate mathematically, numerically, and even grammatically, because depending on the point of view, on the purpose or the context, it can take very different meanings. For instance, one definition could be:

Fairness is the absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics

Once again, this definition is not universal, and as one can see, it is hardly understandable for a computer. A notion of metric has to be defined to know and *measure* to what extent two instances have been fairly treated. Now, the problem is that such a defined metric is arbitrary, and has to be adapted to specific use cases, and the chances are that not everyone will accept it. One of the goals of fairness studies in ML is thus to find a procedure that would lead to the same result, regardless of the metric used.

The following two sections will be respectively dealing with individual fairness, based on [1], and group fairness, based on [2], regarding classification tasks only.

2 Individual fairness study

Individual fairness is the idea that two similar individuals should get similar predictions. The similarity between two individuals is measured thanks to the metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which is an arbitrary positive and symmetric function capturing the distance between 2 instances of the domain set $\mathcal{X}$.

2.1 Definitions

To carry on, let's define some of the required notions mathematically.

Fairness between two instances of the dataset x_1 and x_2, with respect to a distance measure d, and a decision rule (or policy) π, can be seen as the relation provided in [1],

$$|\pi(x) - \pi(x')| \leq d(x, x') + \alpha \tag{1}$$

where $\alpha \in \mathbb{R}_+$ is an hyper-parameter, that is chosen.

The violation is then measured with :

$$v_\alpha(\pi, (x_1, x_2)) = \max(0, |\pi(x) - \pi(x')| - d(x, x') - \alpha) \tag{2}$$

2.2 Fair classification

The principle used by [1] to construct a fair online classifier that is metric free, relies on a reduction to a usual online classifier.

To do so, the algorithm is given at each iteration step t a batch $(\bar{x}^t, \bar{y}^t)$ from the dataset. It deploys the current classifier over it, and spots whether α fairness violations occurred in the predictions.

If so, it picks one of the concerned pairs, and update the batch according to the following rule: **C copies of the first element are made, replacing its current label by 0, and C copies of the second one are made, replacing its current label by 1, and all are added to the batch**.

If not, it picks any element of the batch, and update the batch according to the following rule: **2C copies of the chosen element are made, half of those are labelled 0 and the others are labelled 1**.

Then, the policy rule is updated from all the updated batches from the former steps.

The auditor is given formally by:

$$\mathcal{J} = \begin{cases} \rho = (\rho_1, \rho_2) & \text{if } \exists(\rho_1, \rho_2) : |\pi(x_{\rho_1}) - \pi(x_{\rho_2})| - d(x_{\rho_1}, x_{\rho_2}) - \alpha > 0 \\ null & otherwise \end{cases}$$

The details of the implementation are given in the algorithm 1 from [1].

Intuitively, one may see why in the long run, this leads to a fair classifier. To verify it, loss measurements will be introduced.

```
for $t : 1, ..., T$ do
    Environment chooses the batch $(\bar{x}^t, \bar{y}^t)$
    The batch is audited with $\pi^t$ to get the pair $\rho^t$
    if $\rho^t = (\rho_1^t, \rho_2^t)$ then
        for $i : 1, ..., C$ do
            $x_{k+i}^t = x_{\rho_1^t}^t$ and $y_{k+i}^t = 0$
            $x_{k+C+i}^t = x_{\rho_2^t}^t$ and $y_{k+C+i}^t = 1$
        end
    end
    else
        for $i : 1, ..., C$ do
            $x_{k+i}^t = v$ and $y_{k+i}^t = 0$
            $x_{k+C+i}^t = v$ and $y_{k+C+i}^t = 1$
        end
    end
    $k' = k + 2C$
    $\bar{x}'^t = (x_\tau'^t)_{\tau=1}^{k'}$ and $\bar{y}'^t = (y_\tau'^t)_{\tau=1}^{k'}$
    $\pi^{t+1} = \mathcal{A}_{BATCH}\left((\pi^j, \bar{x}'^j, \bar{y}'^j)_{j=1}^t\right)$
end
```

Algorithm 1: Reduction from Online Fair Batch Classification to Online Batch Classification, from [1]

- The usual Misclassification Loss, given a loss function ℓ over a batch is given by:

$$Err(\pi, (\bar{x}, \bar{y})) = \sum_i \ell(\pi(x_i), y_i) \tag{3}$$

- The α-Fairness Loss is then defined as :

$$Unf_\alpha(\pi, (\bar{x}, \bar{y})) = \begin{cases} 1 & \text{if } (\rho_1, \rho_2) \neq null \\ 0 & otherwise \end{cases} \tag{4}$$

- The Lagrangian loss, combining the two previous ones:

$$\mathcal{L}_{C,\alpha}(\pi, (\bar{x}, \bar{y})) = Err(\pi, (\bar{x}, \bar{y})) + \begin{cases} C(1-\alpha) & \text{if } (\rho_1, \rho_2) \neq null \\ 0 & otherwise \end{cases} \tag{5}$$

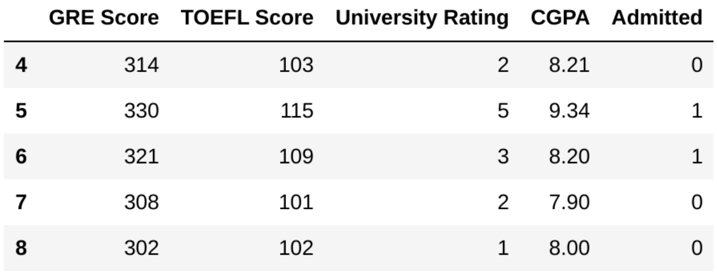
2.3 Implementation of the reduction

The Python code was written on Jupyter notebooks that are available here:

https://github.com/maximeBtn/research_project.git

The experiments were done using LogisticRegression and SGDClassificer from the library *scikit learn*.

The dataset chosen was initially taken from Kaggle (available by clicking here) and is composed of 400 instances. To simplify it, some attributes were dropped, and the probabilities of admission were rounded to 1 when over 0.7 and rounded to 0 otherwise, so the dataset turns into a binary classification. Here's the final shape of the dataset.



	GRE Score	TOEFL Score	University Rating	CGPA	Admitted
4	314	103	2	8.21	0
5	330	115	5	9.34	1
6	321	109	3	8.20	1
7	308	101	2	7.90	0
8	302	102	1	8.00	0

Figure 3: Dataset presentation

The corresponding distributions are given in figure 9.

2.4 Numerical results and conclusion

2.4.1 Offline classification

The best offline classification score with a 5 cross-validation is 0.8475. This will be our reference to all the coming results.

2.4.2 Online classification

One can see on 4 that the score of the classifier converges almost to the one given by the offline classifier. This makes sense, because at some points, the classifier is trained with all the possible instances of our initial dataset.

2.4.3 Reduction from online classification

One can see on 5 that this reduction is not relevant here, whatever the value given to the threshold α. Making copies only disturbs the classification, and therefore doesn't help

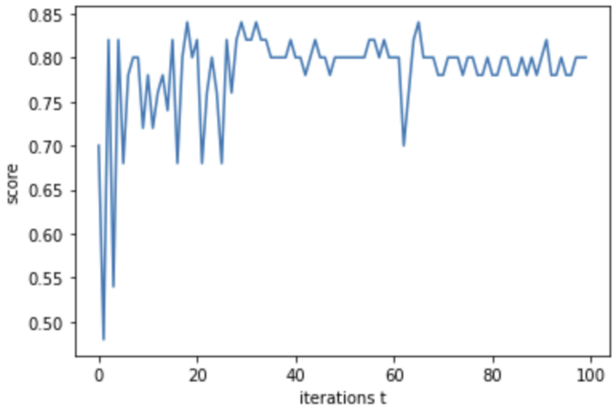


Figure 4: Online classifier score

getting a better fairness over the iterations, it is like a vicious circle.

Therefore, the choice of C, and the copying has to be refined. A statistical sampling is not accurate enough to get a convergence in the training.

3 Group fairness study

To try to solve the problem related to copying in individual fairness, a complementary study on group fairness has been done. To do so, the experiments and approach described below are based on the work and the code developed by the authors of [2].

The idea to get a fair classifier with respect to group fairness is really different. Indeed, the principle relies on optimal transport above all.

3.1 Definitions

Let's express the two main definitions of group fairness when it comes to classification.

Equalized opportunities: A classifier $h(\cdot) : X \to [0, 1]$ satisfies the equal opportunity criterion relative to $\mathbb{Q}$, a given distribution, if

$$\mathbb{Q}(h(X) \geq \tau | A = 1, Y = 1) = \mathbb{Q}(h(X) \geq \tau | A = 0, Y = 1) \tag{6}$$

where τ is the classification threshold.
In other words, two identical instances, deferring only by their sensitive attributes, get positively labelled with the same probability.

Equalized odds: A classifier $h(\cdot) : X \rightarrow [0, 1]$ satisfies the equal odds criterion relative to $\mathbb{Q}$, a given distribution, if

$$\forall y \in \mathcal{Y}, \mathbb{Q}(h(X) \geq \tau | A = 1, Y = y) = \mathbb{Q}(h(X) \geq \tau | A = 0, Y = y) \tag{7}$$

where τ is the classification threshold.
In other words, two identical instances, deferring only by their sensitive attributes, get labelled both positively and negatively with the same probabilities.

3.2 Optimal Transport (OT) to test group fairness

The paper [2] offers a statistical test to verify that a classifier is fair with respect to one of the definitions given above. In our study, only the equalized opportunities approach will be considered.

This test relies on the use of OT, and specifically the use of the type-2 Wasserstein distance.

3.2.1 A quick presentation of OT

Optimal transport is the "cost" of moving a distribution to another. This can be illustrated in 1D with the figure 6.

One can see the source distribution as a pile of dirt, and someone needs to move this pile to somewhere else, and give it another shape, how can he shovel the pile as efficiently as possible so it minimizes his effort ? This is an equivalent to the problem of OT.

3.2.2 Wasserstein distance

This optimal transport can be measured with the following metric, which is the Wasserstein distance.

Wasserstein distance: The type-2 Wasserstein distance between two probability distributions $\mathbb{Q}$ and $\mathbb{Q}'$ supported on Θ is defined as

$$\mathbb{W}(\mathbb{Q}, \mathbb{Q}') = \min_{\pi \in \Pi(\mathbb{Q}, \mathbb{Q}')} \sqrt{\mathbb{E}[c(\epsilon, \epsilon')^2]} \tag{8}$$

where the set $\Pi(\mathbb{Q}, \mathbb{Q}')$ contains all joint distributions of the random vectors $\epsilon \in \Theta$ and $\epsilon' \in \Theta$ under which they respectively have marginal distributions $\mathbb{Q}$ and $\mathbb{Q}'$, and $c : \Theta \times \Theta \rightarrow [0, \infty]$ constitutes a ground metric.

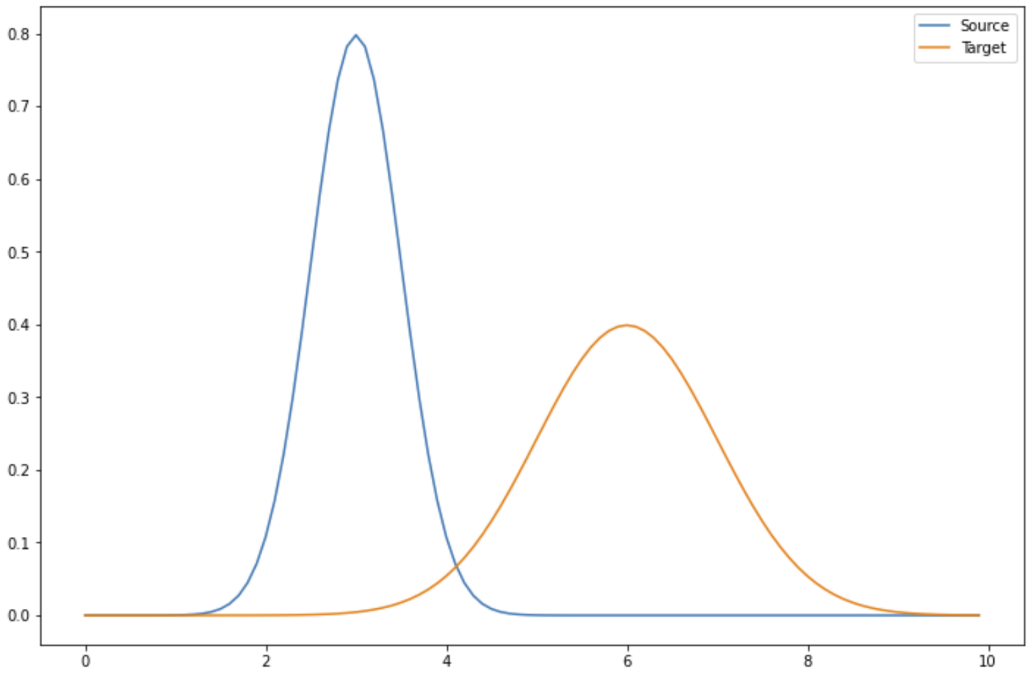


Figure 6: Illustration of Optimal Transport

3.2.3 Principle for testing

The approach adopted by [2], is to measure the Wasserstein distance between the distribution of our data and the closest distribution that would be **fair** with respect to the classifier h used.
Mathematically, the most favorable distribution $\mathbb{Q}^*$ can be written as :

$$\mathbb{Q}^* = \arg\min_{\mathbb{Q} \in \mathcal{F}} \mathbb{W}(\mathbb{P}^N, \mathbb{Q}) \tag{9}$$

where $\mathcal{F}$ is the set of all distributions that are fair with respect to our classifier. Ideally would like $\mathbb{Q}^*$ to be the distribution $\mathbb{P}^N$, which would mean that our distribution is fair.

As shown in [2], for equal opportunities settings, the goal is to compute the **marginal projection of the Wasserstein distance**, as follows:

$$\begin{aligned} \mathcal{R}(\mathbb{P}^N, \hat{p}^N) &= \mathbb{W}(\mathbb{P}^N, \mathbb{Q}^*)^2 \\ &= \sup_{\gamma \in \mathbb{R}} \frac{1}{N} \sum_{i \in \mathcal{I}_1} \min_{k_i \in [0,1/8]} \gamma^2 \lambda_i^2 \|\beta\|_2^2 k_i^2 + \frac{\gamma \lambda_i}{1 + \exp\left(\gamma \lambda_i \|\beta\|_2^2 k_i - \beta^\top \hat{x}_i\right)} \end{aligned} \tag{10}$$

where λ_i is inverse of the marginal probability over (A, Y) associated to the instance of attributes x_i, β is the coefficients of the logistic regression. Once this projection is computed, one can compare the value $N \times \mathcal{R}(\mathbb{P}^N, \hat{p}^N)$ to the value $\eta_{1-\alpha}$ of the limiting distribution θ_{χ^2}, to conclude about the statistical test.

The most favorable distribution $\mathbb{Q}^*$ can be retrieved with:

$$\mathbb{Q}^* = \frac{1}{N}\left(\sum_{i \in \mathcal{I}_0} \delta_{(\hat{x}_i, \hat{a}_i, \hat{y}_i)} + \sum_{i \in \mathcal{I}_1} \delta_{(\hat{x}_i - k_i^* \gamma^* \lambda_i \beta, \hat{a}_i, \hat{y}_i)}\right) \tag{11}$$

where k_i^* and γ^* are the optimum for 10.

3.3 The general idea and the actual implementation

Initially, the purpose of this statistical test was to be the regularizer in our online classification process. Indeed, taking the Wasserstein distance as part of the cost function during the training process should limit the unfairness in the prediction result.

Unfortunately, because of analytical issues with the formulas, it was hard to design a batch gradient descent for the Wasserstein projection.

Therefore, a new idea has been imagined. Considering 2 classifiers. Indeed, one of them, called the *training classifier* is used for the online training, and its accuracy converges to some point, as for an usual online classifier in a mini batch gradient descent. But, at each epoch, the most favorable distribution with respect to the given mini batch is computed. Then, another classifier, called the *fair classifier*, is trained over the fair distribution.

3.4 Results

To make those experiments, the dataset used was the famous COMPAS dataset (related to recidivism rate in the USA), which is known for its unfairness predictions (bias towards colored people), because we needed one with sensitive attributes, here the ethnicity can be used.

As one can see on 7, training on the most favorable distribution did not affect drastically the accuracy of the prediction. There are thus 2 main possibilities. The first one is that the data given was already fair, which is of course wrong, regarding COMPAS dataset. The second one is that our training strategy is not powerful enough.

This second hypothesis can be validated thanks to 8, which clearly shows that the unfairness has almost not changed even though we trained the fair classifier on the most favorable distribution.

4 General conclusion, improvements and personal comments

Altough the several experiments carried out there didn't lead to incredible results, I've learned that implementation of theoretical results is not necessarily an easy thing. One can imagine that the big part of the work is done, but it's not true.

Indeed, some mathematical results can be easy to prove, to identify, but when it comes to practice, one realizes that this is not convenient, because of limited resources, storage, time computation...

I have found this project more than interesting, above all in the last weeks, where I had finally quite a good grasp over the whole topic. The main difficulties for me were definitely diving into the subject, understanding precisely the different notions presented in the paper. The other ones were also related to unexpected results in my implementation (non-convergence of theoretical converging sums for instance...).

What I will remind from this project is that many of the concepts seen in the papers were way beyond my reach when I started to read them in November, and now I've discovered quite a lot of branches of ML that I didn't know of. All of them could potentially give many ideas for PhD subjects... I'm quite proud for having worked on a topic that is so important in ML, even though a bit disappointed by the few results I obtained.

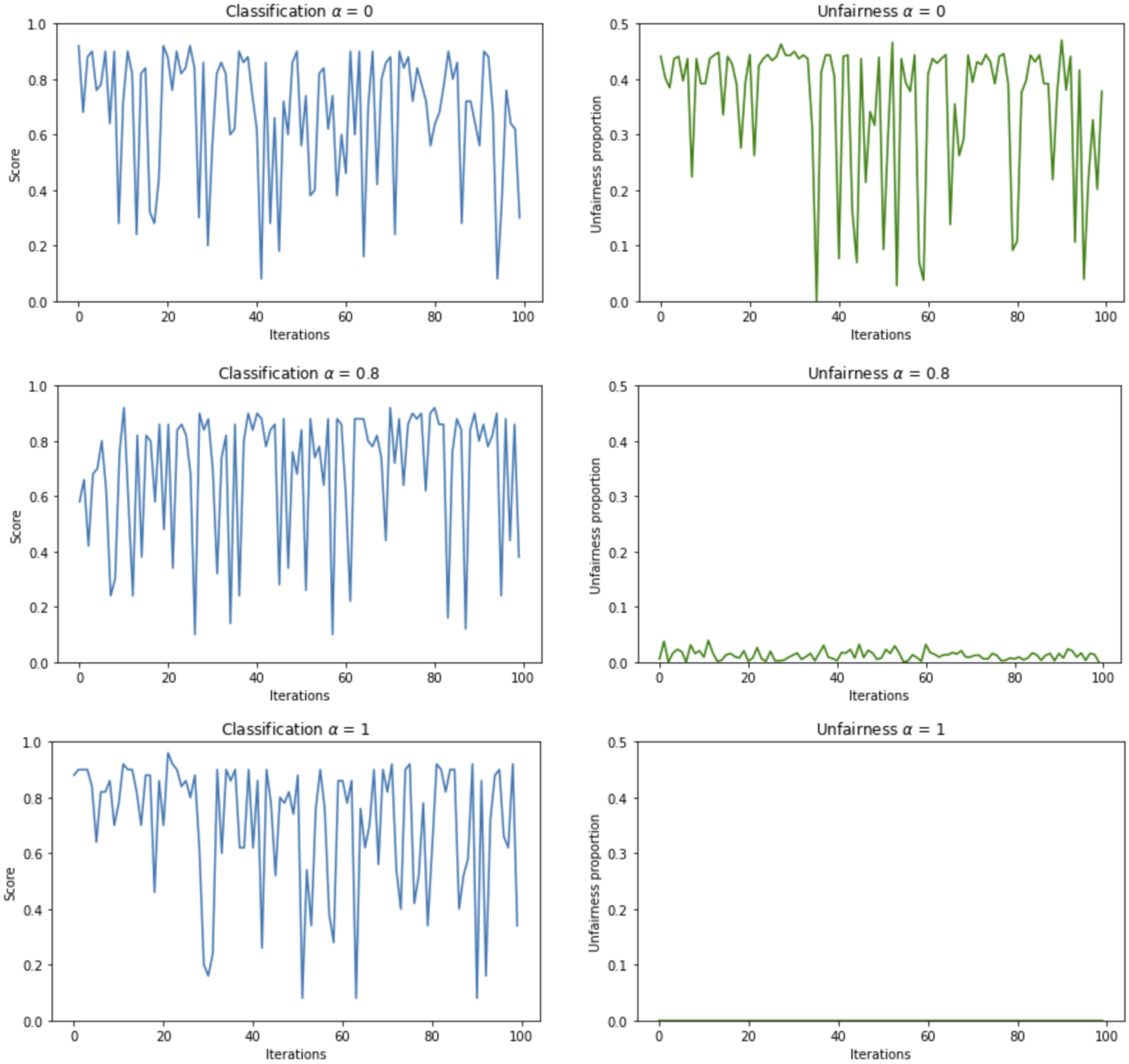


Figure 5: Reduction score

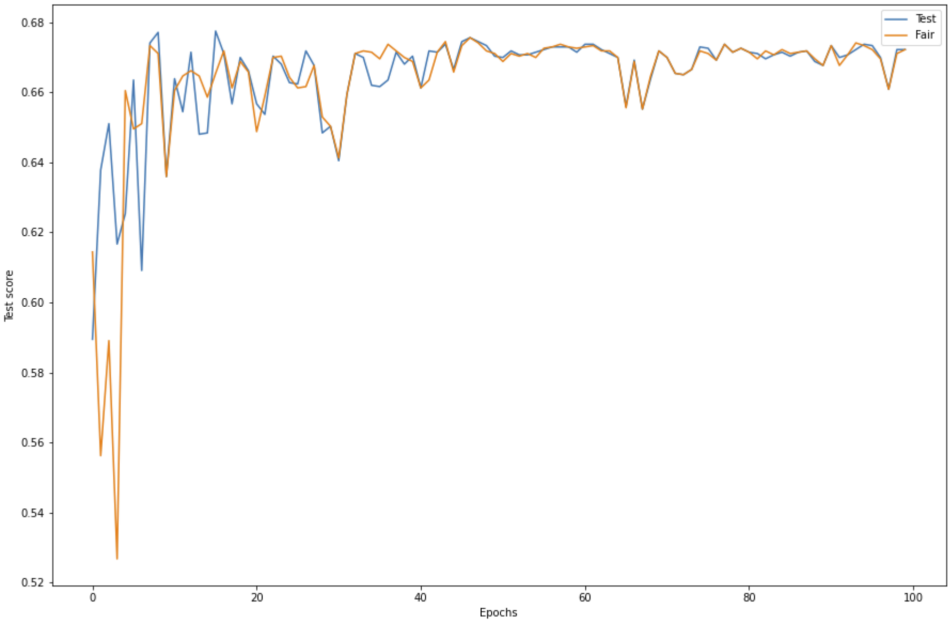


Figure 7: Score comparison between fair and training classifier

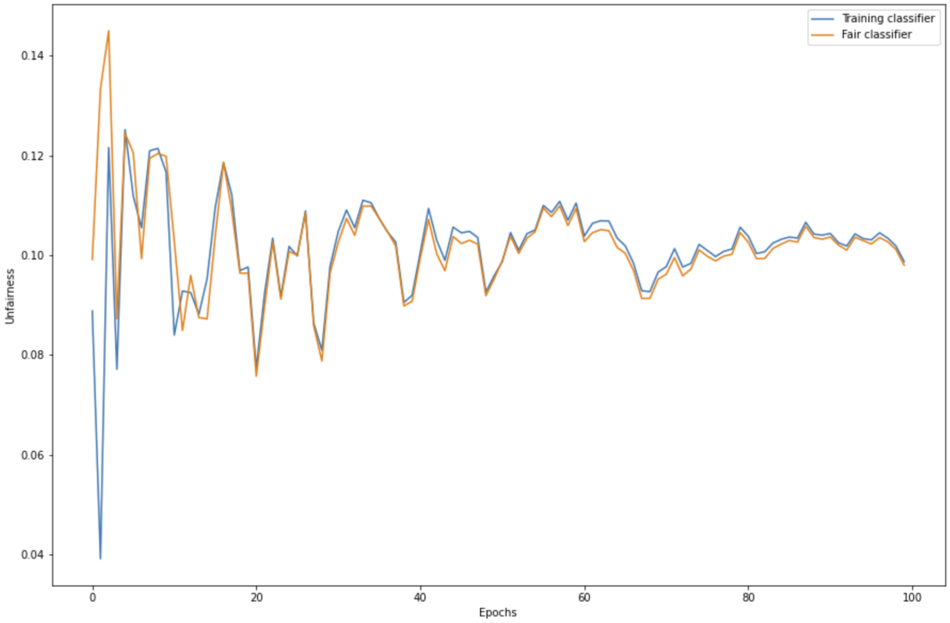


Figure 8: Fairness comparison between fair and training classifier

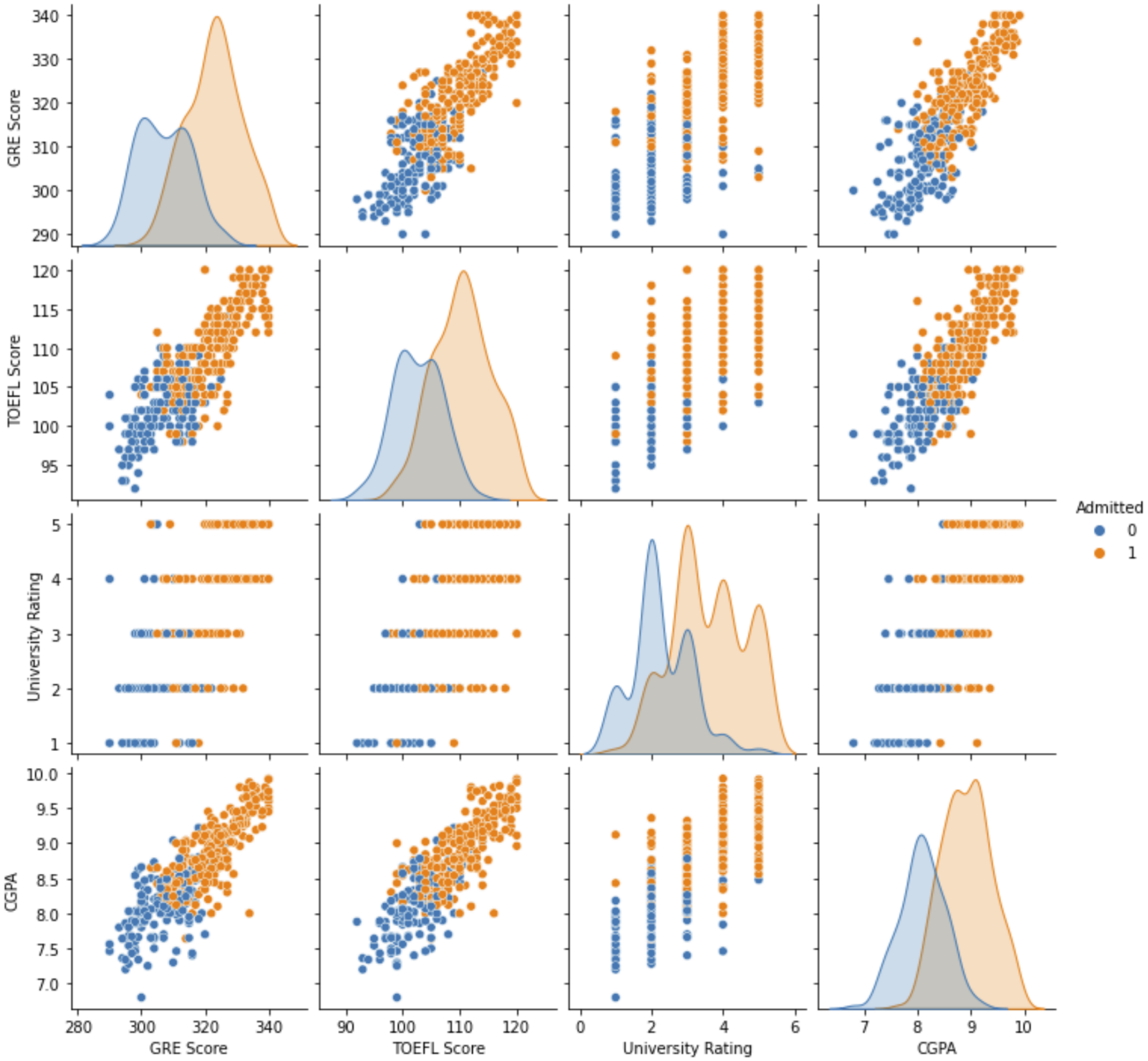


Figure 9: Dataset pair-plots

References

[1] Y. Bechavod, C. Jung, and Z. S. Wu, “Metric-Free Individual Fairness in Online Learning,” *arXiv:2002.05474 [cs, stat]*, Jan. 2021, arXiv: 2002.05474. [Online]. Available: `http://arxiv.org/abs/2002.05474` (visited on 04/05/2021).

[2] B. Taskesen, J. Blanchet, D. Kuhn, and V. A. Nguyen, “A Statistical Test for Probabilistic Fairness,” *arXiv:2012.04800 [cs, stat]*, Dec. 2020, arXiv: 2012.04800. [Online]. Available: `http://arxiv.org/abs/2012.04800` (visited on 04/05/2021).

[3] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A Survey on Bias and Fairness in Machine Learning,” *arXiv:1908.09635 [cs]*, Sep. 2019, arXiv: 1908.09635. [Online]. Available: `http://arxiv.org/abs/1908.09635` (visited on 04/05/2021).