Parametrised Data Sampling for Fairness Optimisation

Vladimiro G. Zelaya
Newcastle University
Newcastle upon Tyne, UK
c.v.gonzalez-zelaya2@ncl.ac.uk

Paolo Missier
Newcastle University
Newcastle upon Tyne, UK
paolo.missier@ncl.ac.uk

Dennis Prangle
Newcastle University
Newcastle upon Tyne, UK
dennis.prangle@ncl.ac.uk

ABSTRACT

Improving machine learning models' fairness is an active research topic. Most approaches to it focus on a particular fairness definition. We propose a parametrised training-set-resampling method, which allows optimising in both a fairness-definition and classification-model agnostic manner. Given a binary protected attribute and a binary label, we correct the positive rate for both the favoured and unfavoured groups through four different resampling methods. Three fairness definitions are adjusted to ratio forms allowing us to measure a classifier's level of fairness. Results of experiments over three public benchmark datasets (Census Income, COMPAS and German Credit) are presented, with the ultimate goal of optimising the correction level for a particular dataset and fairness definition.

ACM Reference Format:
Vladimiro G. Zelaya, Paolo Missier, and Dennis Prangle. 2019. Parametrised Data Sampling for Fairness Optimisation. In *Proceedings of Explainable AI for Fairness, Accountability & Transparency Workshop (KDD XAI).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The increasing presence of automated decisions in our lives has led to a rising concern about the way in which these decisions are taken. The main area on which research has focused is *fairness*. Although we may have an intuitive idea of what this concept means, trying to formally define it has proven to be troublesome, and many different definitions of fairness have arisen [13]. A problem with this is the well-known fact that many of these definitions are incompatible with each other: a decision rule satisfying one of the definitions may well prove to be very unfair for a different one [4]. For example, determining university admissions through gender quotas may achieve demographic parity, but make the acceptance rates for good students of different genders disparate. We propose a fairness-definition-agnostic method to optimise a classifier's behaviour with respect to a particular fairness definition, without incurring a big loss in predictive power. One of our proposed method's variations results in a generalisation of Kamiran and Calders' *Preferential Sampling* [9].

Fairness-aware machine learning is defined by Friedler et al. [7] as preprocessing techniques modifying input data so that any classifier trained on said data will be fair. According to [10], there are four main ways in which to make appropriate adjustments to data in order to enforce fairness: suppressing certain features, also known as *fairness through unawareness* [8], *massaging* variable values [3], reweighing features [12] and resampling data instances [15], [9], [16]; our proposed method belongs to this last category.

Part of the inspiration for parametrising the level of correction to apply came from the idea of *worldviews* by Friedler et al. [6], referring to the set of assumptions made about the *construct*, the metric space including all the relevant features to a decision task. In their paper, Friedler et al. [6] introduce three distinct worldviews: *what you see is what you get* (WYSIWYG), *we are all equal* (WAAE) and *structural bias*. WYSIWYG refers to the assumption of the construct space being essentially the same as the observed space, i.e. all the available data. WAAE, on the other hand, assumes that all groups of individuals with respect to a potentially discriminatory (or *protected*) attribute should perform equally well regarding the classification task. Finally, structural bias assumes that there is a larger distortion between groups than there is between individuals when mapping between the observed space and the construct space, i.e. protected attribute groups may play a more relevant role in the decision rule than they actually should. This calls for a modified decision rule that corrects this distortion, or taking *affirmative action* (AA).

The main idea is to correct a classifier's unfair behaviour through training-set resampling. We propose a method to modulate these corrections via a continuous parameter d, which will produce corrected positive and negative ratios for different protected attribute groups, and these new ratios will in turn be enforced by a correcting function, based on a resampling method.

According to Berk [1], one of the main problems with data resampling is the loss of prediction accuracy provoked by such interventions. Our method has two responses to that: on one side, even at the most severe correction levels, the loss in accuracy for most of our sampling methods is relatively low. On the other hand, this loss in accuracy may also be controlled through parameter d, allowing for a decision in the amount of accuracy for fairness trade-off the user is willing to accept.

Our fairness correcting method permits enforcing a particular worldview by selecting the adequate value for its correction parameter. A second use for our correction parameter is to optimise a classifier's predictions with respect to a particular fairness definition, by only applying as much correction as needed. This is the main contribution of this paper, as it provides a fairness-definition and classifier-agnostic method to generate fair decisions through a simple and efficient data resampling procedure. Finally, we show how a classifier may be optimised for fairness without much accuracy loss in the process.

KDD XAI, 5 August 2019, Anchorage, Alaska, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 METHODOLOGY

This initial version of our method is meant to be used for correcting training datasets for classification tasks. We have focused on datasets with both binary protected attributes and labels. For some cases, this could be addressed by making the protected attribute binary by grouping together labels which are not relevant for separate analysis.

2.1 Definitions

Across this paper, we will be constantly referring to concepts that might have a fairness-specific connotation. Hence, we present our meaning for them next.

We will say a binary classifier's label can be *positive* or *negative* referring to the desirable and non-desirable outcome of a prediction, respectively. For example, in a classification task deciding whether to grant a user a bank loan, the positive label would refer to getting the loan, and the negative one to being rejected.

The *protected attribute* (PA) refers to a variable in data that may be an object of discrimination, due to historical bias or otherwise. In our particular case we will be dealing with *binary* PAs, meaning there will only be two PA groups, every instance in the data belonging to one of those.

We will call the ratio of the number of positive instances divided by the total number of instances in a specific group the *positive rate*, or PR of the group. Analogously, the ratio of the number of negative instances in a group divided by the total number of elements in the group will be referred to as the *negative rate* (NR) of the group.

Among the two PA groups, the one having the *highest* PR will be referred to as the *favoured* (F) group, while the other one will be referred to as the *unfavoured* (U) group.

2.2 Parametrising Correction

We introduce the *disparity correction* parameter d, which may be used for two different objectives:

(1) To enforce a particular *worldview* [6], as defined above.
(2) To optimise a classifier with respect to a particular fairness definition.

This parameter may take values $d \in [-1, 1]$, affecting the amount of correction going into the training set. A d-value of 1 is associated with the WYSIWYG worldview, leaving the training set as-is. A d-value of 0 is associated with WAAE, making the PRs for both the favoured and the unfavoured groups equal with the population PR. Finally, a d-value of -1 is associated with AA, as it makes the PR of the favoured group equal with the unfavoured group's original PR and viceversa.

Our main objective, though, is not to enforce a specific worldview by preprocessing, but to be able to optimise a classifier's predictions with respect to a fairness definition. In order to do so, we propose to train classifiers associated with different d-values, evaluate the fairness metric of interest over the obtained models' predictions and select the d-value optimising such a metric.

Given a dataset D, let A be a binary PA of D, with unfavoured group U and favoured group F with respect to their PR in the training set, let X be the set of unprotected attributes of D and let Y be the binary label, taking values 0 and 1. The way our correcting parameter works is as follows.

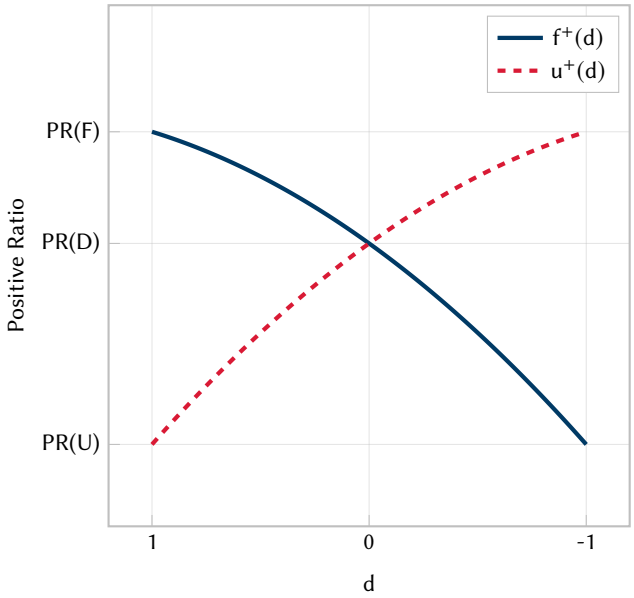

Figure 1: PR-correcting functions for favoured and unfavoured groups. The d-axis is reversed for interpretability.

Given $d \in [-1, 1]$, we wish to find a function $f^+(d)$ that will yield a corrected PR for F, such that:

$$f^+(1) = \mathrm{PR}(F), \quad f^+(0) = \mathrm{PR}(D), \quad f^+(-1) = \mathrm{PR}(U).$$

The simplest smooth function satisfying these conditions is a quadratic polynomial. The intention of our correcting method is to increase the PR of U while decreasing the PR of F. At $d = 0$, we wanted both groups to have the same PR (reflecting the *WAAE* worldview), Likewise, at $d = -1$ we wish the PR for both groups to be completely reversed. For this reason, we also define $u^+(d) = f^+(-d)$, the corrected PR of U. The algebraic forms of these two polynomials are:

$$f^+(d) = ad^2 + bd + c$$
$$u^+(d) = ad^2 - bd + c,$$

and solving for a, b and c given the desired constraints results in coefficients:

$$a = \frac{\mathrm{PR}(F) + \mathrm{PR}(U)}{2} - \mathrm{PR}(D), \quad b = \frac{\mathrm{PR}(F) - \mathrm{PR}(U)}{2}, \quad c = \mathrm{PR}(D).$$

Plots for both PR-correcting functions are shown in Figure 1. In this and all subsequent figures, the d-axis is reversed for better interpretability.

Corrected NR for both groups will simply be

$$f^-(d) = 1 - f^+(d) \quad \text{and} \quad u^-(d) = 1 - u^+(d).$$

We use these corrected ratios $f^+(d), f^-(d), u^+(d)$ and $u^-(d)$ to produce a d-resampled training set which, when used to fit a classifier, will have an effect on the different fairnesses of its predictions.

2.3 Sampling Strategies

We implemented four different correcting functions, each based on one of the following sampling strategies, in order to achieve the desired PRs for both the favoured and unfavoured populations:

Random Undersampling (Under). Favoured positive and unfavoured negative subgroups are randomly undersampled, i.e. individuals are removed randomly until the favoured positive and unfavoured negative groups get to the necessary size.

Random Oversampling (Over). Favoured negative and unfavoured positive groups are randomly oversampled, i.e. existing individuals in the relevant groups get replicated until the favoured negative and unfavoured positive groups get to the necessary size.

SMOTE Oversampling (SMOTE). Favoured negative and unfavoured positive groups are oversampled using the *Synthetic Minority Oversampling TEchnique* (SMOTE) [2]. Similar to random oversampling, in this case synthetic datapoints are generated based on a k-nearest-neighbours algorithm. Again, the number of generated datapoints will depend on the necessary size for the favoured negative and unfavoured positive groups.

Preferential Sampling (PS). Introduced by Kamiran and Calders in [9, 10]. Ranker is learnt from training data and used to sort every subgroup by positive-class probability. Favoured positive and unfavoured negative subgroups are undersampled, taking out datapoints with highest/lowest probabilities, respectively. Favoured negative and unfavoured positive groups are randomly oversampled from as many copies as needed of the sorted data, adding in datapoints with lowest/highest probabilities, respectively. For comparison purposes, we will refer to the original, unparametrised version of PS as PS_0.

Once the training set has been rebalanced with our method, the resulting classifier learnt from the corrected training set should see a change in the fairness of its predictions, in the sense that the ratios obtained from equations 1, 2 and 3 should change their value with respect to the corresponding uncorrected ratio.

A final step in finding the optimal correction for a specific fairness definition is to compare the resulting fairness ratios for different values of d, in order to select the one that produces the ratio closest to 1. As we will see on Section 3.1, it is usually easy to find d-values close to the optimal, but for different fairness definitions the optimal d-value will usually be different too.

2.4 Fairness Definitions

In this work we analyse how our proposed method can improve classifier $\hat{Y}$ predictions with respect to three fairness definitions, defined in this section. These definitions are presented in [13].

Demographic Parity. The probability of being classified as positive class is the same across PA subgroups:

$$P\left(\hat{Y}=1 \mid A=U\right)=P\left(\hat{Y}=1 \mid A=F\right). \tag{1}$$

Equality of Opportunity. The probability of being classified as positive class for true positives is the same across PA subgroups.

$$P\left(\hat{Y}=1 \mid A=U, Y=1\right)=P\left(\hat{Y}=1 \mid A=F, Y=1\right). \tag{2}$$

Counterfactual Fairness. A classifier $\hat{Y}$ is counterfactually fair if under any context $X=x$ and $A=a$,

$$P\left(\hat{Y}_{A\leftarrow U}=1 \mid X=x, A=a\right)=P\left(\hat{Y}_{A\leftarrow F}=1 \mid X=x, A=a\right), \tag{3}$$

where $\hat{Y}_{A\leftarrow U}$ and $\hat{Y}_{A\leftarrow F}$ denote the value of $\hat{Y}$ had A taken the values U or F, respectively.

Even though these definitions are clear, in order to measure how fair a classifier's decisions are, it is more convenient to think of quotients associated with these. We can define a classifier's *demographic parity ratio* (*DPR*) through the following equation, closely connected to equation 1:

$$DPR=\frac{P\left(\hat{Y}=1 \mid A=U\right)}{P\left(\hat{Y}=1 \mid A=F\right)}. \tag{4}$$

Evidently a classifier's predictions will satisfy demographic parity if and only if $DPR=1$ on equation 4. Also, even though $DPR=1$ may not be achieved, the closer *DPR* is to 1, the fairer the classifier will be, with respect to demographic parity.

We define the *equality of opportunity ratio* (*EOR*) analogously to *DPR* as:

$$EOR=\frac{P\left(\hat{Y}=1 \mid A=U, Y=1\right)}{P\left(\hat{Y}=1 \mid A=F, Y=1\right)}. \tag{5}$$

Again, the closer this ratio is to 1, the fairer a classifier will be, this time with respect to Equality of Opportunity.

In the same fashion as *DPR* and *EOR*, we may define a classifier's *counterfactual fairness ratio* (*CFR*) as:

$$CFR=\frac{P\left(\hat{Y}_{A\leftarrow U}=1 \mid X=x, A=a\right)}{P\left(\hat{Y}_{A\leftarrow F}=1 \mid X=x, A=a\right)}. \tag{6}$$

In practice, evaluating these probabilities *for any context* $X=x$ *and* $A=a$ can be troublesome, since that would imply evaluating this ratio on every combination of X and A present in T.

As proxy to this, we follow the procedure shown in Figure 2: we intervene test set T twice, assigning every individual in T the PA values U and F and obtaining $T_{A\leftarrow U}$ and $T_{A\leftarrow F}$ as results, respectively. This approach was defined by Kilbertus et al. [11] as *proxy fairness*. We then define CFR^* as the ratio of the PRs of predictions for both intervened test sets:

$$CFR^*=\frac{\text{PR}(T_{A\leftarrow U})}{\text{PR}(T_{A\leftarrow F})}. \tag{7}$$

The same as with *DPR* and *EOR*, the closer CFR^* is to 1, the fairer our classifier will be.

3 EXPERIMENTS

We tested our methods over three datasets commonly used on machine learning fairness research literature: Census Income (Income) [5], COMPAS [14] and German Credit (Credit) [5]. A summary of the main features of each dataset may be seen in Table 1.

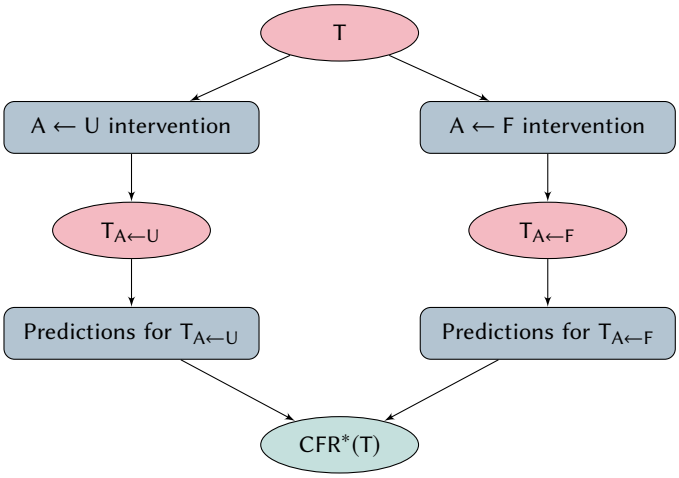

Figure 2: Calculating CFR* for test set T.

Table 1: Datasets used for our experiments.

Dataset	Instances	PA	Favoured	Positive Class
Income	48842	Sex	Male	Over $50k income
COMPAS	6907	Race	White	Will not recidivate
Credit	1000	Gender	Male	Will repay loan

For every dataset, we performed the following experiment 50 times, and then averaged the results for stronger statistical soundness:

(1) Random train/test split the data with 90/10 proportion.
(2) For counterfactual-fairness checking, make two copies of the *test* set T and intervene A as either U or F, obtaining $T_{A\leftarrow U}$ and $T_{A\leftarrow F}$, respectively, as shown in Figure 2.
(3) For each of the four sampling functions obtain 11 different training sets, each corresponding to a different value of $d \in \{1, 0.8, 0.6, \ldots, -1\}$.
(4) For each of these training sets, fit a logistic regression (LR) model.
(5) For every model, get predictions for T, $T_{A\leftarrow U}$ and $T_{A\leftarrow F}$.
(6) Compute metrics for accuracy, demographic parity, counterfactual fairness and equality of opportunity, as well as the coefficients for every feature in the model.

We then proceeded to analyse the resulting fairness metrics, and compared our results with PS_0 (which is essentially the same as applying PS with $d = 0$ using our method).

3.1 Results

As expected, disparity correction has an effect on a classifier's PR. Figure 3 shows a particular instance of this, using random undersampling as our correcting method, for the COMPAS dataset. As may be seen, at the optimal d-value—the point at which the curves cross over—both the F and U groups achieve the same PR, which is the population PR. This is unlike PS_0, for which F and U PR never change and are quite far from each other and from the population PR.

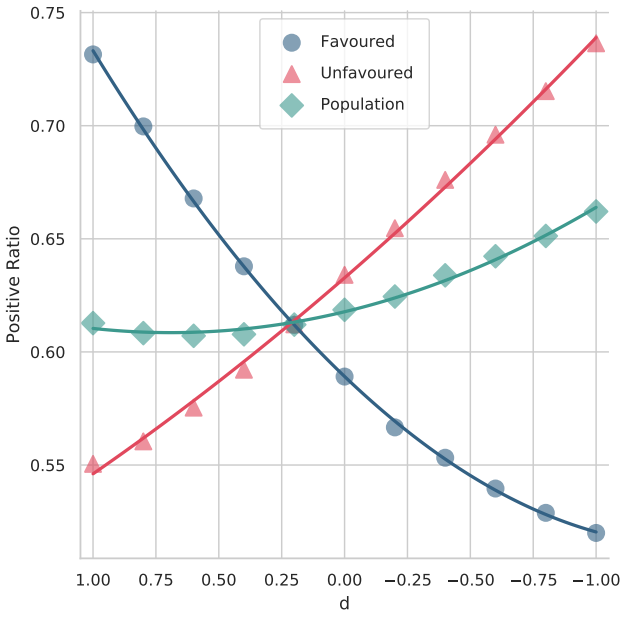

Figure 3: COMPAS PR by undersampling correction.

Figure 5 shows the resulting metrics for accuracy, *DPR*, *EOR* and *CFR** for all three analysed datasets. It is worth noting that both the Credit—and to a lesser extent COMPAS—datasets have a relatively smaller number of instances, hence the trends appearing in the figures for said datasets are not as smooth as the ones for the Income dataset.

3.1.1 Accuracy. In all three datasets, accuracy decreased monotonically with correction increases, as may be seen in the top row of Figure 5. It is worth noting that the decrease in accuracy resulting from increased correction was not particularly severe, with less than a 4% accuracy drop on all the non-PS methods for every dataset even at the highest correction level. Whether this trade-off is beneficial or not will ultimately be application-specific.

3.1.2 Fairness Ratios. For all the analysed fairness ratios, correction had a much stronger effect when using PS as our resampling method. This is explained partially by the fact that PS resamples all four population subgroups at once, while the other three methods only resample two of said groups. This stronger effect of PS has both a positive and a negative consequence. The positive one is that, in general, a smaller d-value will be required to achieve optimal fairness ratios. The negative consequence, though, is the higher variability in fairness ratios due to the change in the value of d. This means that optimising for a specific fairness ratio may greatly magnify another ratio's unfairness.

For both the Income and Credit datasets, PS_0 performed quite well regarding *DPR*, getting ratios close to the optimal. For *EOR* and *CFR**, however, PS_0 overcorrects by a big margin on every dataset.

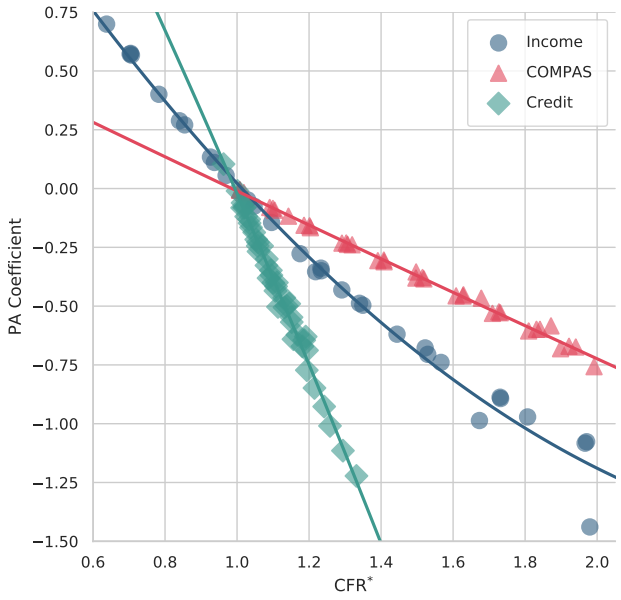

Figure 4: CFR* by PA coefficient for all three datasets, with quadratic-regression curves for each one.

In every dataset, all sampling methods achieved optimal *DPR*, *EOR* and *CFR**, but did so with different d-values. Steep rates of change in PS, though, mean that a d-value only slightly away from the optimal can cause *DPR* to vary greatly. Interestingly, *DPR* optimality was the hardest to enforce. For all non-PS methods (Under, Over and SMOTE), it took d-values close to -1 to achieve optimal *DPR*.

Another interesting trend that may be observed in Figure 5 is that, given d_{DPR}, d_{EOR} and d_{CFR^*}—the optimal d-values for *DPR*, *EOR* and *CFR** under a particular correction method—it roughly follows that

$$d_{DPR} \leq d_{EOR} \leq d_{CFR^*},$$

i.e. achieving demographic parity will require greater correction than is needed for equality of opportunity, which in turn requires greater correction than counterfactual fairness.

3.1.3 PA Coefficients. Scatter plotting *CFR** vs. the PA coefficient of the 132 produced model-fits—obtained from fitting eleven d-values over four correction strategies for three datasets—revealed an interesting relation between these two quantities: the fit regression line for each dataset's scatter passes through point $(1, 0)$, as may be seen in Figure 4. This makes perfect sense, as the coefficient for a feature being 0 means that the feature is completely irrelevant for the model (and its predictions). Hence interventions $A \leftarrow U$ and $A \leftarrow F$ over a datapoint will be totally indistinguishable to the model. In other words, a LR model with PA coefficient 0 will always be proxy fair. On the other hand, we did not find any such relation for *DPR* and *EOR* with respect to the *PA* coefficient.

4 CONCLUSION

In this paper, we have defined a parametrised fairness optimisation method that is both fairness-definition and classification-model agnostic, which may be used to enforce a particular worldview. By using correcting functions based on training set resampling, we have shown that our method produces fairness-optimal predictions with a small loss in predictive power. For future work directions, we intend to analyse our method's performance on different fairness definitions from the ones presented in this work. Another interesting and relevant research direction is to extend our method to work on non-binary PA and Label cases. We also intend on further improving our data resampling methods, in order to minimise the accuracy loss. Finally, a method for optimising over several fairness definitions at once may also be worth working on.

ACKNOWLEDGMENTS

This work was produced as part of the EPSRC Centre for Doctoral Training in Cloud Computing for Big Data programme. Vladimiro G. Zelaya would like to thank the financial support provided by Universidad Panamericana, Mexico and the Digital Institute at Newcastle University, UK.

REFERENCES

[1] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.
[2] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN 10769757.
[3] Silvia Chiappa and Thomas PS Gillam. Path-specific counterfactual fairness. *arXiv preprint arXiv:1802.08139*, 2018.
[4] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.
[5] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.
[6] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.
[7] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 329–338. ACM, 2019.
[8] Pratik Gajane and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.
[9] Faisal Kamiran and Toon Calders. Classification with no discrimination by preferential sampling. In *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*, pages 1–6. Citeseer, 2010.
[10] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.
[11] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*, pages 656–666, 2017.
[12] Emmanouil Krasanakis, Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, and Yiannis Kompatsiaris. Adaptive sensitive reweighting to mitigate bias in fairness-aware classification. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 853–862. International World Wide Web Conferences Steering Committee, 2018.
[13] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076, 2017.
[14] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9, 2016.
[15] Donald B Rubin. The use of matched sampling and regression adjustment to remove bias in observational studies. *Biometrics*, pages 185–203, 1973.
[16] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. Capuchin: Causal database repair for algorithmic fairness. *arXiv preprint arXiv:1902.08283*, 2019.

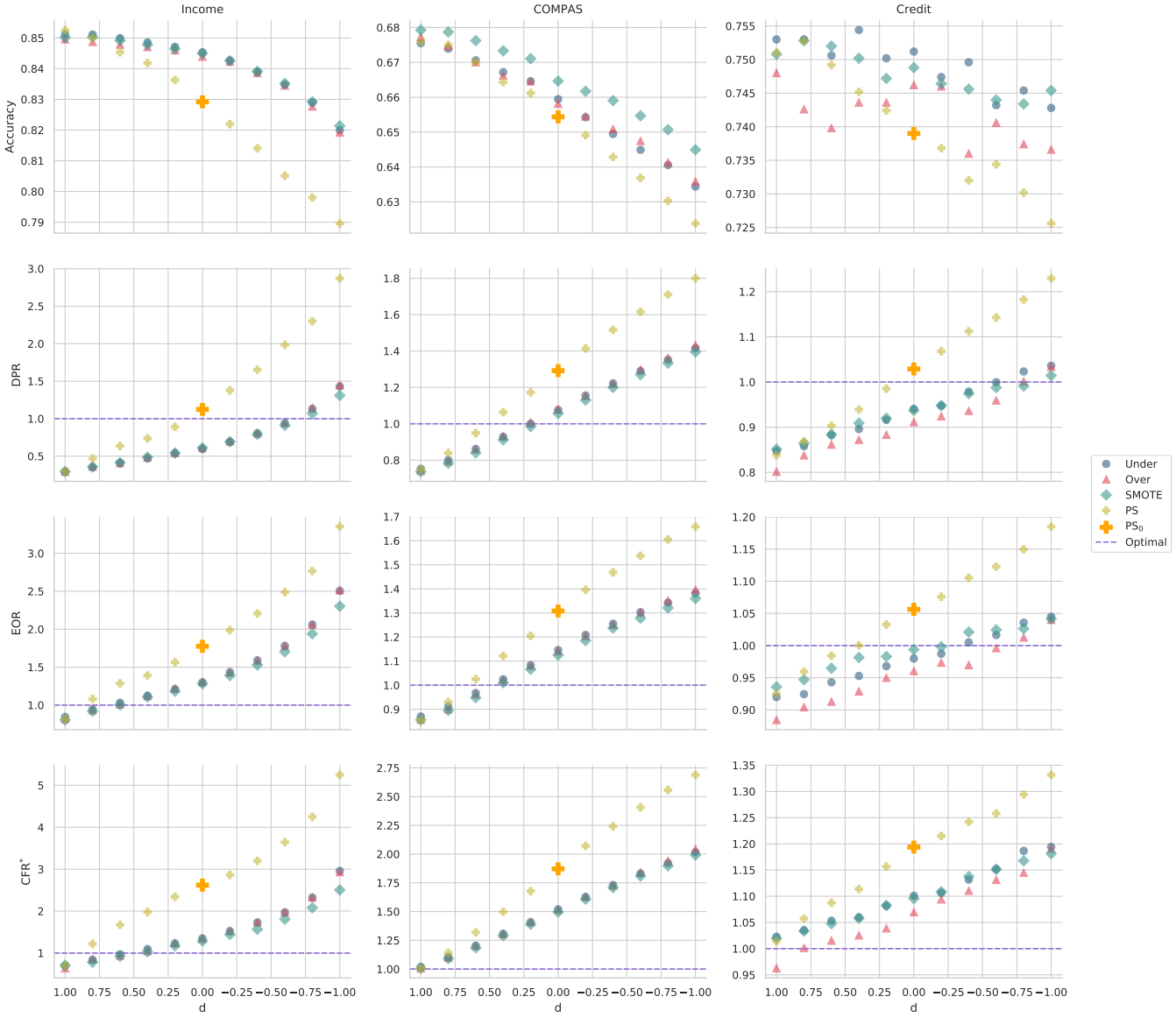

Figure 5: Accuracy and fairness ratios by correction level d for all three datasets.

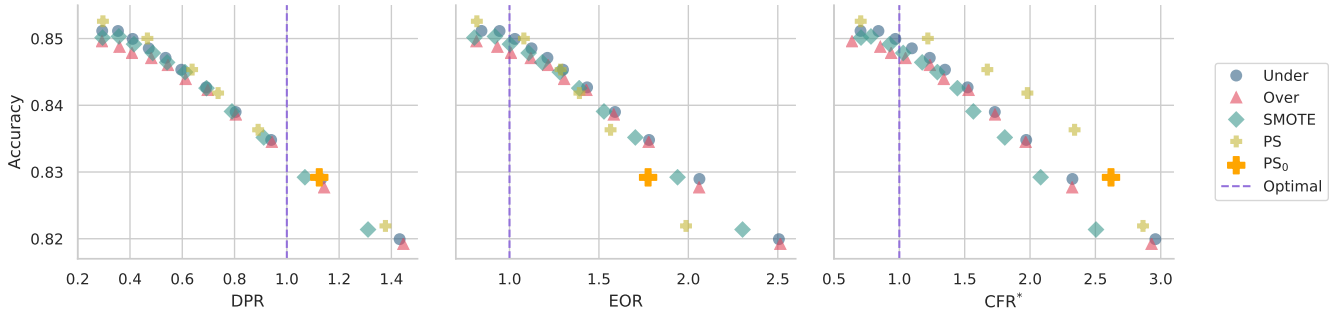

Figure 6: Accuracy by DPR, EOR and CFR^* for the Income dataset.

A REPRODUCIBILITY SUPPLEMENT

A.1 GitHub Repository

Our repository includes both of the scripts used for our analyses, as well as all of the datasets in both their original and cleaned-up versions. It may be found at:

https://github.com/vladoxNCL/fairCorrect

A.2 Software Requirements

Our algorithms were written and run in Jupyter Notebooks 4.4.0 over a Python 3.6.5 kernel.

The following Python 3 packages need to be installed for our notebooks to work properly:

- pandas 0.23.0
- NumPy 1.14.3
- scikit-learn 0.19.1
- imbalanced-learn 0.3.3
- Matplotlib 2.2.2
- Seaborn 0.9.0

A.3 Scripts

There are two notebooks in our repository:

- **data_cleanup.ipynb**: Helper notebook, used to convert the original data files into a clean and one-hot encoded version, suitable for data analysis.
- **fairCorrect.ipynb**: Our main notebook.
  - In the first half, the correction algorithms are coded. The user needs to specify the name of the desired dataset the scripts will be run over in the second code block, by setting the `dset` variable to the appropriate string value (`Income` by default).
  - The second half generates most of the figures in the paper. Some of the generated plots are dataset-specific, hence the `dset` variable should be set according to the dataset to be analysed. For both scripts, all the *savefile* commands have been commented out, as the savepath needs to be specified by the user.

A.4 Data

We have performed our analyses on three fairness, accountancy and transparency benchmark datasets: Census Income, COMPAS and German Credit. The original CSV files may be obtained from the following URLs, as well as from our repository:

- **Census Income**
  https://archive.ics.uci.edu/ml/datasets/census+income
- **COMPAS**
  https://github.com/propublica/compas-analysis
- **German Credit**
  https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

A.5 Algorithm

Pseudocode for our *fairCorrect* Under, Over and SMOTE correction methods is shown in algorithm 1. All three methods work in a very similar way, with the only differences being the groups to resample (fPos and uNeg for Under, fNeg and uPos for Over and SMOTE) and the sampling function itself.

Algorithm 1: *fairCorrect* data correction method.

Data:
T: a training set with binary PA A and binary label Y,
$d \in [-1, 1]$: the correction parameter,
$s \in$ {Under, Over, SMOTE}: the sampling method.
Result:
Tcorr: a d-corrected training set.

```
/* Get favoured F and unfavoured U datasets    */
for i in {0, 1} do
    T_i = T[A == i];
    Tpos_i = T[(Y == 1) and (A == i)];
    PR_i = size(Tpos_i) / size(T_i);
end
j = argmax_{i∈{0,1}}(PR_i);
F = T_j;
U = T - F;
/* Get positive rates for U, F and T           */
for d in {U, F, T} do
    PR_d = size(d[Y == 1]) / size(d);
end
/* Correcting polynomials                      */
a = ((PR_F + PR_U) / 2) - PR_T;
b = (PR_F - PR_U) / 2;
c = PR_T;
fpr = a * d**2 + b * d + c;
upr = a * d**2 - b * d + c;
/* Split F and U into Positive and Negative    */
[fPos, fNeg] = [F[Y == 1], F[Y == 0]];
[uPos, uNeg] = [U[Y == 1], U[Y == 0]];
/* Random undersampling case                   */
if s == Under then
    /* Correct fPos and uNeg groups            */
    f_k = fpr / (1 - fpr);
    u_k = (1 - upr) / upr;
    fPosSize = f_k * size(fNeg);
    uNegSize = u_k * size(uPos);
    fPos = undersample(fPos, size=fPosSize);
    uNeg = undersample(uNeg, size=uNegSize);
/* Random oversampling and SMOTE cases         */
else
    /* Correct fNeg and uPos groups            */
    f_k = (1 - fpr) / fpr;
    u_k = upr / (1 - upr);
    fNegSize = f_k * size(fPos);
    uPosSize = u_k * size(uNeg);
    fNeg = oversample(fNeg, size=fNegSize);
    uPos = oversample(uPos, size=uPosSize);
end
/* Get all four groups back together           */
Tcorr = concat(fPos, fNeg, uPos, uNeg);
```