Fairness by Thresholding?

A performance evaluation of thresholding and in-processing

Master's Thesis
submitted to

Prof. Dr. Stefan Lessmann

Humboldt-Universität zu Berlin
School of Business and Economics
Chair of Information Systems

by Janek Willeke (556286)

in partial fulfillment of the requirements for the degree M.Sc

Berlin, November 05, 2019

Contents

Abstract

Fair machine learning has the potential to mitigate some of the problems posed by algorithmic decision making. There are countless algorithms that jointly optimize for accuracy and low disparate impact, one of the most common fairness criteria. It is unclear, however, how this approach compares to a simple post-processing by thresholding of a classifier that was optimized for accuracy only. Such an approach would not only be easier to implement but also more flexible to use, providing a good baseline for classifiers reducing impact disparities. This Master's thesis relates thresholded classifiers and in-processing classifiers in how they reduce disparate impact from a theoretical perspective and empirically evaluates their performance on five data sets.

1 Introduction

Machine learning is increasingly applied to inform consequential decisions that impact people's lifes in fields as diverse as credit rating, criminal justice and medicine. Yet decisions in these fields have ethical consequences and even legal constraints, making it paramount to study the fairness of the algorithms applied. Consider the case of an algorithm Amazon used to rate job candidates. The algorithm learned that maleness is a defining quality of an Amazon employee (Dastin, 2018). It rated applicants lower hailing from an all women's college or applicants mentioning words related to female activities such as women's club in their resume. There are countless more examples of how algorithms influence decision making in the real world and might lead to unfair decisions being taken, see e.g. Mehrabi et al. (2019) for a short overview. Developing algorithms aiming for predictive quality but also constrained by moral considerations has thus recently become a focus of research. Many different notions of fairness have been proposed along with many different criteria to measure whether and how strongly algorithms violate these norms.

One criteria of fairness, called demographic parity, imposes that the algorithm's decision should be independent of membership to a certain sensitive group, e.g. gender or ethnicity. Disparate impact is the most popular way to measure how strongly a classifier deviates from this (Barocas et al., 2017)(Zafar et al., 2017b). It measures by how much the positive rates of a classifier differ across sensitive groups by taking the ratio of the positive rates. While there exist a range of algorithms jointly optimizing for fairness and accuracy, so called in-processing classifiers, it has recently been questioned if such algorithms are even necessary to achieve fairness (Kleinberg et al., 2018). Instead, a simpler and more versatile solution would be to compute different prediction thresholds for each sensitive groups, such that the positive rates are equalized. This would also be provably optimal in the case of a classifier that learns the true ranking of class probabilities. In no realistic application of fair machine learning can it be assumed that this is the case, however. This master's thesis aims to empirically evaluate how post-processing by thresholding performs vis-à-vis in-processing algorithms. Additionally, it aims to isolate characteristics of the classifiers

that might cause disparities in performance.

First, section 2 will give a broad overview of the fair machine learning literature and introduce the fairness notion underlying the classifiers tested in this thesis, demographic parity, in more detail. Section 3 gives an overview of algorithms that mitigate impact disparities. The following subsection does in some length describe by what constraints the in-processing classifiers tested in this thesis reduce impact disparities. The aim is to identify how they differ from the constraints imposed by thresholding a classifier. The post-processing proposed by this thesis is then presented in section 3.2. Section 4 attempts to relate classifier constraints to generalization properties by considering their effect on the joint distribution of the margins and the sensitive variable. Margins here are taken to mean the signed distances to the decision boundary in the input space. Section 5 presents the methodology and empirical results. The appendix contains a short overview of the notation and definitions of the most important terms used in this master's thesis. Throughout the text, the reader will be referred to the appendix for more detailed treatments of some of the topics. Code for all models and replications is provided with the thesis.

2 Notions of Fairness

Fairness in machine learning commonly refers to the absence of discrimination. Discrimination can be broadly defined as actions, practices or policies that disadvantage one socially salient group relative to a relevant comparison group (Altman, 2016). However, this leaves ample space for interpretation as to what constitutes a disadvantage. Different fairness criteria in machine learning can be based on different notions of what constitutes such a disadvantage or different ways of measuring disadvantage. The two notions of fairness most prominently held in the literature are equality of odds and it's offshoot equality of opportunity and demographic parity. Equality of odds constrains the error rates of classifiers to be equal across groups (Hardt et al., 2016). The idea is that when unbiased data is available a classifier should strive to give a prediction that is as accurate as possible. Under this notion of what constitutes a disadvantage, if e.g. the algorithm predicts a higher likelihood of re-offence for whites than for blacks than this would not constitute discrimination if it reflects the true likelihood. Equality of opportunity (Hardt et al., 2016) focuses solely on the false positive rate. Unless there is a trade-off in accuracy across groups, equality of odds would entail decreasing the prediction quality for one group. It remains to be seen how much enforcing equality of odds leads to better predictions for the worse-classified group instead of just worse predictions for the better-classified group. Some recent works have actively tried to increase the predictive quality for worse-off group, consider e.g. (Dwork et al., 2018).

There exist a wide array of other fairness notions, e.g. individual fairness (Dwork et al., 2012), where fairness is defined as similar treatment for similar individuals, with the simi-

larity measure hard to determine. Preference-based notions of fairness (Zafar et al., 2017a), where algorithms are constrained by some notion of group preferences and discrimination is allowed have recently gained some traction. While one might want to enforce all or some these fairness criteria jointly, this is generally not possible (Kleinberg et al., 2016). It is quite intuitive that when e.g. the empirical distribution $P_{sample}(Y = 1|X, s)$ differs between sensitive groups that equality of odds and demographic parity are in conflict. In order to mitigate algorithmic discrimination membership to a sensitive group needs to be taken into account (Dwork et al., 2012), as otherwise discrimination could be introduced via variables correlated with the sensitive variable.

Disparate impact

In many cases, however, it is reasonable to assume that the labels reflect a historical bias. An example is the aforementioned hiring algorithm Amazon used. As it's decision was based on past hiring decisions, it is hard to see how human prejudice did not influence hiring and therefore the data. Often it is reasonable to assume that the label only depends on sensitive group membership through these biases. To reflect reality, predictions should be made independent of sensitive group membership. The fairness criterion derived from this observation is disparate impact, defined for sensitive variable s and predicted labels $\hat{y} \in \{-1, 1\}$ as:

$$di = \frac{min(P(\hat{y} = 1, s = 0), P(\hat{y} = 1, s = 1))}{max(P(\hat{y} = 1, s = 0), P(\hat{y} = 1, s = 1))}$$

For this thesis, if positive rates are unequal and 0 for one group, define the disparate impact di as 0 and if positive rates are 0 for both group define di as 1. The classifier achieves demographic parity when $di = 1$. The p-rule $di \leqslant p$ presents a looser fairness criterion, where p is taken to be some smaller value, 0.8 is mentioned most often and derived from US law (Zafar et al., 2017b). Demographic parity has been considered in a wide array of works in the machine learning literature, some of which will be discussed in the next section. There has, however, also been criticism of how demographic parity might impact people's wellbeing. Liu et al. (2018) can show that under certain conditions classifiers reducing disparate impact can harm the disadvantaged group, by e.g. giving a loan to people that are very likely to default. For this, however, they assume that the data is completely unbiased. It remains to be seen when and how often mitigating impact disparities can harm the disadvantaged group. There exists for example research on mismatch in school admission that points to students from disadvantaged minorities admitted in the context of affirmative action overachieving relative to their application's strength (Dale and Krueger, 2014) and (Melguizo, 2008). Relating such empirical work to how algorithms reducing impact disparities would affect people's lives is an important understudied topic.

3 Algorithms that reduce impact disparities

Algorithms mitigating impact disparities fall into three categories: Pre-processing, in-processing and post-processing. It is possible to pre-process a data set in order to erase

information on the sensitive variable or reweight the data points, as proposed by e.g. Kamiran and Calders (2012) or Calmon et al. (2017). In a similar vein, both Sattigeri et al. (2018) and Xu et al. (2018) propose GANs to learn a fair data distribution. This debiased data can then be fed to a classifier that optimizes for accuracy only. The idea is that if the pre-processing can purge the data of all information on the sensitive variable, a classifier trained on this data cannot learn predictions that have disparate impact.

3.1 In-processing classifiers

Most research instead focuses on devising classifiers optimizing for accuracy while being constrained to lessen impact disparities. Intuitively, this seems an easier task, as only the predicted labels have to be independent of the sensitive group. Kamishima et al. (2012) implement a penalized logistic regression. Zafar et al. (2017b) and Celis et al. (2019) propose convex constraints, allowing convex optimization of the risk function. Agarwal et al. (2018) iteratively manipulate the sample weights such that a classifier nears demographic parity. Edwards and Storkey (2015) first use an adversary to reduce disparate impact, Zhang et al. (2018), Madras et al. (2018), Raff and Sylvester (2018) and Adel et al. (2019) all propose adversarially constrained classifiers. Louizos et al. (2015) introduce a multi-level variational autoencoder with a maximum mean discrepancy (Gretton et al., 2007) penalty that drives the algorithm towards less disparate predictions. Botros and Tomczak (2018) modify the architecture by replacing the gaussian prior with a vamp prior and replacing the MMD penalty with a penalty on the mutual information between the sensitive variable and the latent variable.

This section introduces in more detail the in-processing classifiers that will later on be tested against post-processing by thresholding. A prerequisite for explaining possible performance differences is to clearly identify how the classifiers reduce impact disparities, which the following subsections will do. Classifiers from the literature were selected based on their architecture. Fair logistic regression and fair neural network architectures are considered, as in-processing classifiers and thresholded classifiers should be of the same model type to allow for direct performance comparison. Thresholding requires margins, making these models obvious candidates. Post-processing could of course also be extended to e.g. support vector machines, but this is beyond the scope of this thesis. By including both linear and non-linear models, it might be possible to e.g. uncover if thresholding performs better on neural network that more accurately approximate $P(y|X,s)$. The implementation of Kamishima et al. (2012) is taken from the AIF360 module (Bellamy et al., 2018). For Zafar et al. (2017b) Python 2 code is available via Github but had to be adjusted for Python 3. The adversarial models and the variational fair autoencoder were implemented by the author in Pytorch. Replication results can be found in the appendix.

3.1.1 Constrained Logistic Regression

Zafar et al. (2017b) implement a constrained fair logistic regression. They approximate a p-rule constraint by constraining the empirical covariance of the margins, i.e. the signed distances to the decision boundary $X'\theta$ where θ are the weights with the sensitive variable s. The new constrained optimization problem can then by linearity be reduced to:

$$min_\theta \sum_{i=1}^{N} -log(P(y_i|x_i,\theta)) \; s.t.$$

- $\sum_{i=1}^{N}(s_i - \bar{s})\theta' x_i \leqslant c$
- $\sum_{i=1}^{N}(s_i - \bar{s})\theta' x_i \geqslant -c$

This is a convex program and can be solved in polynomial time [1]. Optimization is done via sequential quadratic programming. Constraining the covariance to be 0 entails constraining the group wise sums $\sum^{i:s=j}\theta' x_{s_j i}$ to be equal, since s is binary. Note that the constraint might have the effect of not only equalizing the sum of margins across sensitive groups, but also leading to margins smaller in magnitude. This is because the difference in sum of margins accross sensitive groups depends on the magnitude of the margins. As the constraint enforces only equality of the sums of margins to the decision boundary across sensitive groups, e.g. diverging variance might lead $C(y|X,s) > 0.5$ to differ across groups. Consider the extreme case where $P(\theta' x_{s_0} = -0.1) = 1$, $P(\theta' x_{s_1} = 0.1) = 0.5$ and $P(\theta' x_{s_1} = -0.3) = 0.5$. Then the mean of $\theta' x$ is equal for the sensitive groups. Yet it holds that $P(C(X, s = 0) > 0.5) = 0$, but $P(C(X, s = 1) > 0.5) = 0.5$. This shows that constraining the sums of margins to be equal may not prevent disparate impact.

Kamishima et al. (2012) like Zafar et al. (2017b) implement fair logistic regression by regularization. They introduce a term to the logistic regression loss function that penalizes a rough approximation of mutual information between $C(y|X,s)$ and s, defined as the Kl-divergence between the joint distribution and the product of the marginal distribution. The KL-divergence is then estimated over the empirical samples and added to the loss function: $\sum_{x_i,s_i}\sum_{y\in 0,1} C(y_i|X_i,s_i) log(\hat{C}(y|s_i)/\hat{C}(y_i))$. Here $\hat{C}(y|s_i) = C(y|\hat{x}_{s_i}, s_i)$ and to estimate $C(y)$ s is marginalized out. This avoids marginalizing out X, which Kamishima et al. (2012) argue might be too computationally intense.

It is unclear how far this estimator of mutual information proposed by Kamishima et al. (2012) is from the true value. $C(y|\bar{x}_{s_i})$,i.e. replacing each x_i with the average for every sensitive group, can be very different from $C(\bar{y}|x,s_i)$, the average predicted y value for a given s which results from marginalization. It is more reasonable to think of the Kamishima et al. (2012) penalty as punishing differences in the sum of predicted class probabilities.

[1] the problem can be extended to other convex margin based classifieres, such as svm

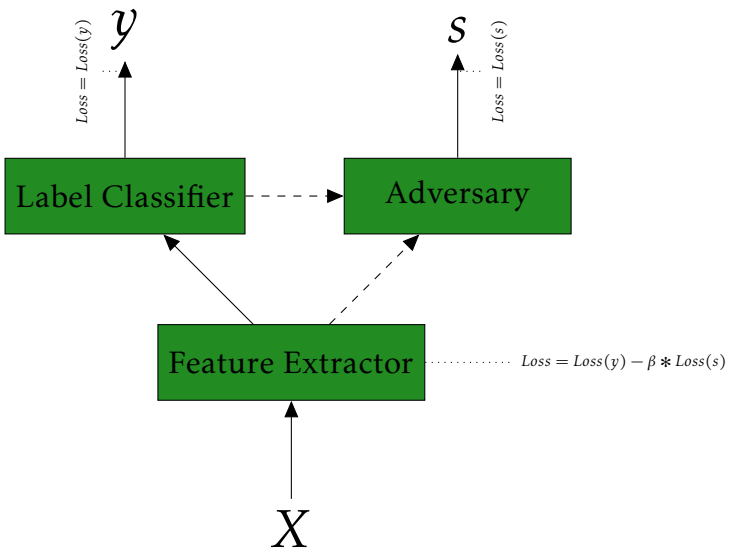

Figure 1: A diagramm of adversarial model architectures.

Green are neural networks. Dotted edges are either/or, i.e. the features extractor is optional, with the adversary predicting the sensitive variable from the margins.

Unless $C(y|\bar{x}_{s_i})$ is equal across sensitive groups, the predicted class probabilities for one sensitive group will be weighted negatively by the log term, the predicted class probabilities for the other sensitive group positively. Again, as was the case for the penalty propoed by Zafar et al. (2017b), the Kamishima et al. (2012) penalty might be zero even when the positive rates differ by a great deal across sensitive groups. Optimzation is done via the conjugate gradient method. The penalty is non-convex, which might lead to the algorithm getting stuck in bad local minima.

3.1.2 Adversarially constrained classifiers

Most non-linear fair in-processing algorithms are based on neural network architectures. Fair adversarial models include Edwards and Storkey (2015), Beutel et al. (2017), Zhang et al. (2018), Madras et al. (2018) and Raff and Sylvester (2018). Fair adversarial networks draw from a seminal paper by Ganin and Lempitsky (2014) on domain adaptation. The procedure Ganin and Lempitsky (2014) propose trains the model on the known labels, while aiming to prevent an adversary from being able to predict the domain from the highest level representation of the features. In order to induce the domain predictor to predict the correct domain label, the gradient is reversed after the adversary's output weights. Otherwise it would just learn to flip the labels. While this approach bears some similarity to generative adversarial networks (Goodfellow et al., 2014) there is one crucial difference. While the only information the generator of a GAN has about the true labels comes channeled through the adversary, the label predictor of the domain adaptation network as proposed by Ganin and Lempitsky (2014) has access to the ground truth.

Edwards and Storkey (2015) first adapt domain adversarial learning to fair classification. They change the model proposed by Ganin and Lempitsky (2014) by slightly varying the learning rate of the adversary relative to the label classifier. Weight updates for the adversary and label classifier are alternated every batch. The Edwards and Storkey (2015) architecture and learning procedure was otherwise implemented as described and will be one of the models evaluated. Zhang et al. (2018) modify the architecture by giving the adversary access only to the predicted logits [2]. This is how an adversarially constrained logistic regression was implemented for this master's thesis.

3.1.3 Variational fair autoencoder

Louizos et al. (2015) propose a hierachical latent variable model where the latent variables independent of the sensitive variable. An additional MMD-penalty further encourages independence.

Variational autoencoders (Kingma and Welling, 2013) are latent variable models learned via the reparameterization trick where functions between latent variables are neural networks. The variational autoencoder can be extended to the (semi-) supervised case by introducing a hierachy of latent variables, as introduced in Kingma et al. (2014). Louizos et al. (2015) build on this model by adding a sensitive variable s which is assumed to be independent of the latent variables z_1, z_2. This gives three latent variables z_1, z_2, y where the encoders $Q(z_1|X,s), Q(z_2|z_1,y)$ and decoders $P(z_2), P(z_1|z_2,y)$ are multivariate Gaussians and the distribution of y and X depends on the task at hand. The evidence lower bound for y observed [3] is:

$$ELBO = \sum^{N} E_{Q(z_{1i}|X_i,s_i)}[-KL(Q(z_{2i}|z_{1i},y_i)||P(z_2)) + logP(X_i|z_{2i},s_i)]$$
$$+E_{Q(z_{1i}|X_i,s_i)Q(z_{2i}|z_{1i},y_i)}[logP(z_{1i}|z_{2i},y_i) - log(Q(z_{1i}|X_i,s_i)]$$

by the same logic as the standard variational autoencoder.
If y were independent of s the latent variables should not be strongly dependent on s, as all information on s gets injected back into the latent variable before reconstruction and carrying unnecessary information is penalized. However, in the case that is of interest, where y is dependent on s, the latent variables would still carry information on s. Therefore, an MMD (Gretton et al., 2007) penalty on $Q(z_1|X,s)$ is introduced. $MMD(x,y) = K(x,x) + K(y,y) - 2 * K(x,y)$ computes differences in feature means by the kernel trick, where $MMD(x,y) = 0$ iff $P(x) = P(y)$. Therefore introducing a penalty $\beta * MMD(Q(z_1|X,s=0), Q(z_1|X,s=1))$ penalizes dependence of the latent variable z_1 on s.

[2] They also propose changing the gradient calculation by subtracting the linear projection of the gradient w.r.t. the classifier loss onto the gradient w.r.t. the adversary's loss. This prevent the gradient from changing in the direction of the adversary. This did not improve results in the experiments run for this master's thesis and was therefore not implemented.
[3] For the unsupervised case y values can be imputed, see the paper for details

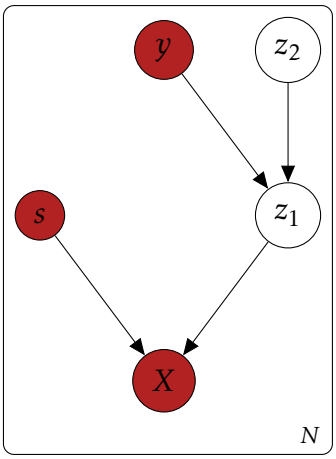

Figure 2: A diagramm of the VFAE generative model

maroon are observed variables

Both, the MMD penalty for the variational fair autoencoder and the negative adversarial loss in neural network based adversarial models have the effect of encouraging independence of the transformed feature distribution in the penultimate layer and s, and thereby also the signed distance to the decision boundary in that layer. This is obvious for the MMD penalty, as it penalizes any dependencies between s and the margin distribution $m(x,s)$. For the adversarial constrained models, any dependency between s and the transformed features (neural networks) or the margins (logistic regression) will also be penalized by the negative adversarial loss, as it's classification accuracy is obviously to a large extent determined by this dependency. It should be expected that for the neural networks, this independence carries into the distances to the decision boundary in the input space. These are in general much stronger constraint than only equalizing sums of the margins across sensitive groups. If sensitive groups have a similar distributions except for different means, the constraints proposed by Zafar et al. (2017b) and Kamishima et al. (2012) should also have the effect of enforcing independence between the margin distribution and s. Thresholding only constrains the cdfs of the margins of the sensitive groups to be equal at two points, i.e. $P(m(x,s=0) < \sigma^{-1}(t_{s_0})) = P(m(x,s=1) < \sigma^{-1}(t_{s_1}))$. In section 4 it will be discussed how these constraints might impact generalization.

3.2 Post-processing

Research on post-processing algorithms is limited. Lipton et al. (2018) only compare fair classifiers without access to the sensitive attribute at prediction time to post-processed classifiers with access to the sensitive variable. Here, the post-processing classifier outperforms the classifiers by Zafar et al. (2017b) and Kamishima et al. (2012). Kleinberg et al. (2018) proofs that given an estimator that learns an optimal ranking of likelihoods and given that the sensitive variable is binary, post-processing by thresholding allows optimal fair predictions. If the rank-ordering of likelihoods learned by the classifier reflects the true likelihoods, i.e. for a classifier C and it's estimated conditional density, $C(y_i|x_i,s_i) > C(y_j|x_j,s_j)$ if and only if $P(y_i|x_i,s_i) > P(y_j|x_j,s_j)\ \forall i,j$, one can always construct

a fair classifier by first splitting the data by sensitive groups. Then equal positive rates are the results of setting the k_{s_0}, k_{s_1} highest ranking observations above group specific thresholds t_{s_0}, t_{s_1} to positive for the groups s_0, s_1, with k_{s_0}, k_{s_1} representing equal proportions of each sensitive group. Yet it cannot be assumed that it is possible to learn the optimal ranking of likelihoods in any realistic use case of fair machine learning even if we have access to the true labels. If the data is biased as is assumed in most applications where impact disparities are to be removed, it is almost impossible. As such, it is unclear how far post-processing by thresholding deviates from the optimal fair decision, if at all.

3.2.1 Finding the optimal thresholds on training data

The procedure described by Kleinberg et al. (2018) does not explain how to find the optimal thresholds given that the algorithm does not learn the true ranking of likelihoods. In this section, the thesis proposes a post-processing by thresholding that allows trading-off accuracy for a desired level of fairness. The aim is to operationalize and extend ideas from Kleinberg et al. (2018). This means finding the best thresholding on data in order to be able to compare thresholding at different fairness-accuracy trade-offs to in-processing classifiers. Post-processing shifts the decision boundary for every level of the sensitive variable such that the fairness conditions hold, by changing the group thresholds t_s such that $P(C(X,s) > t_s)$ is independent of s. Ordering the data points by their predicted probability and sensitive group, the number of false positives for one sensitive group or false negatives for the other sensitive groups is increased depending on the number of data points in each sensitive group. This is how Lipton et al. (2018) construct their thresholding algorithm. For a classifier that does not capture the true ranking of data points, however, the true labels need to be taken into account. That is because the classifier might commit systematic errors, e.g. have different accuracies across groups. By taking into account the true labels, it is possible to find the threshold that maximize accuracy for the given decision boundary and given a required ratio of positive rates. This master's thesis introduces a post-processing that takes into account the true labels. The optimal group thresholds t_{s_0}, t_{s_1} under the constraint that $di < p$ and under a possible missranking of the probabilities can then be found by a simple integer linear program. To set up the problem, order the per-group label vector y_s, with elements $y_{si} \in 1, -1$ by the predicted probabilities $C(y_i|x_i, s_i)$. Let $w \in \{0$, be the decision variable that is 1 whenever a y_i should be set to positive, 0 when not. The objective to be maximized is the dot product of the decision variable and the true labels $w'y$. Here

$$w'y = (w_{s_0}, w_{s_1})'(y_{s_0}, y_{s_1})$$

i.e. the group separate rank-ordered vectors of predicted probabilities are simply joined and their dot product with the decision variables presents the objective. The feasible region is described by:

- $w_{s_i1} \geqslant w_{s_i2} \geqslant w_{s_i3}$... for $i = 0, 1$ (n-2 equations)
- $\sum_{j=1}^{n_{s_0}} \frac{w_{s_0j}}{n_{s_0}} - \sum_{j=1}^{n_{s_1}} \frac{w_{s_1j}}{n_{s_1}} * p \geqslant 0$
- $\sum_{j=1}^{n_{s_1}} \frac{w_{s_1j}}{n_{s_1}} - \sum_{j=1}^{n_{s_0}} \frac{w_{s_0j}}{n_{s_0}} * p \geqslant 0$
- $w \in \{0, 1\}$

Where w_{s_i1} is the decision variable for the data point in group s_i with the highest predicted probability. The objective maximizes accuracy. It amounts to summing up the true positives and substracting the false positives. It suffices to count the data points classified as positive, as underestimating the amount of positive samples would lead to a suboptimal results and thus further iterations. The group specific thresholds t_s then only needs to be set to the predicted probabilities of the lowest ranking $w_{si} = 1$. The first constraints ensure that all data points below the threshold are classified as negative, all above as positive. The second and third constraints constrain the thresholded classifier to follow the p-rule.
This procedure allows not only finding the optimal thresholds that achieves perfect fairness, but allows trading off a pre-specified amount of disparate impact for accuracy. The algorithm can choose not to trade of all the allowed fairness, if this increases accuracy, thereby finding the optimal threshold satisfying the p-rule. A performance comparison to Lipton et al. (2018) can be found in the appendix. While integer linear programs can in general be exponential-time algorithms, computing an optimal threshold for predictions on the largest data set adult took less than two minutes. It is unclear how the algorithm would scale to much larger data sets.

3.2.2 Performance limits of thresholding

While this yields the optimal threshold for a given classifier's margins and a desired level of fairness, in general it cannot be assumed that post-processing by thresholding yields the optimal fairness-accuracy trade-off for the data the classifier is trained on. The optimum for a given fairness-accuracy threshold might not be attainable when optimzing for accuracy first and thresholding after. There will be cases when the thresholded classifier attains this optimum, however. If the classifier learns the true ranking of the likelihoods, thresholding gives optimal results (Kleinberg et al., 2018). This is not even necessary to achieve perfect performance on the sample. It is quite obvious that if the classifier's capacity is large enough, it can correctly classify all data points in the sample no matter how complicated the relationship between features and labels [4]. To achieve the perfect fairness-accuracy trade-off, it then just needs to flip some predictions. Yet, to limit generalization error, one will try to limit the classifier's capacity (Vapnik, 2013). A limited classifier first optimizing for accuracy might get stuck in a maximum optimal for accuracy, but suboptimal for performance in the sense of the fairness-accuracy trade-off. The appendix provides an example.

[4] unless there exists label noise

4 Fairness generalization

Yet even if thresholding achieves better within-sample performance, it might not generalize due to the weak constraint post-processing puts on a classifier. In this section, a connection to existing literature on generalization and the margin distribution is made. Specifically, it will be investigated how properties of the joint distribution of the margins and the sensitive variables might improve fairness generalization. This thesis defines margins as the signed distances to the decision boundary. Classifiers that impose strong constraints on this distribution might generalize better if there exist such properties. To the best of the author's knowledge, no previous research considers this connection. There exists almost no research on fairness generalization. Friedler et al. (2019) show that fairness varies strongly across data splits, but do not attempt to explain this. Huang and Vishnoi (2019) propose a classifier that is more stable when trained on slightly varying data. This is different from fair generalization, however. Non-convex models might find very different minima even when retraining on the same data, yet still generalize well. Cotter et al. (2018) propose a classifier with a critic monitoring fairness on generalization data. They do not attempt to explain the generealization gap, however.

Classical statistical learning established the link between a classifier's capacity and it's margin's width (Vapnik, 1999). A classifier's capacity in turn upper bounds it's generalization error (Vapnik, 2013). Recent literature shows that large margins in the input space improve a classifier's generalization independently of it's capacity, compare e.g. (Jiang et al., 2018). If the margin $m(X_i, s_i)$ of a classifier in the input space is γ at a point (x_i, s_i), then for any disturbance (δ_x, δ_s) with $||(\delta_x, \delta_s)||_2 < ||\gamma||^2$ it holds that (Elsayed et al., 2018):

$$sign(m(x_i, s_i)) = sign(m(x_i + \delta_x, s_i + \delta_s))$$

Such a disturbance could e.g. be noise due to sampling, assuming that points in a new sample are drawn near points of an old sample. The link to generalization of disparate impact is direct. If a classifier's predictions are not flipped by e.g. sample noise, then di will remain unchanged.

Any $y_i \neq \hat{y}_i$ decreases accuracy. Flipping labels in equal proportions for sensitive groups will not change disparate impact, however. If the margin distribution around the decision boundary in the input space of a classifier were independent of s, it's fairness should be more robust to noisy resampling. Consider the extreme case of flipping all labels around the decision boundary. To formalize, for a distance γ to the decision boundary db and predicted labels $\hat{y} \in \{-1, 1\}$,

$$\Delta di = di - di(\hat{y}l), \quad l_i = \begin{cases} -1 & \text{if } |m(x_i, s_i)| \leqslant \gamma \\ 1, & \text{otherwise} \end{cases}$$

If the disparate impact of a classifier only changes slightly under such strong perturbations, it could be expected to be more robust to changes due to sampling variation. For any classifier with the margin distribution independent of s, Δdi would be 0. This immediately follows from independence as

$$P(m(x,s) \in \{+\}, s \in \{1\}) = P(m(x,s) \in \{+\})P(S \in \{1\})$$

and for points classified as negative

$$P(m(x,s) \in \{-\}, S \in \{1\}) = P(m(x,s) \in \{-\})P(S \in \{1\})$$

i.e. proportions of s would be equal. Here $\{+\}$ denotes all points classified as positive within γ of the margin. In this case, the classifier's disparate impact remains unchanged if labels are flipped for all points close to the decision boundary. Therefore, one would assume that a classifier that enforces independence between predictions and the sensitive variable is more robust to changes in the sample distribution, if they are also independent in the input space. In general, for a thresholded classifier that achieves demographic parity it only holds that $P(C(y|X, s = 0) > t_0) = P(C(y|X, s = 1) > t_1)$, where t_i denotes the group specific threshold. As such, proportions of points near the margin could be arbitrarily different. Therefore it should be less robust to differences due to resampling.

Margins

To test these intuitions on data, distance to the decision boundary, i.e. the margins first need to be defined. For a logistic regression the euclidean distance to the decision boundary in the input space is simply $|\theta'(x_i, s_i)| = |\sigma^{-1}C(y_i|x_i, s_i)|$, where θ are the weights and σ the sigmoid function. For neural networks, $\sigma^{-1}C(y_i|x_i, s_i)$ only gives the distance to the decision boundary in the penultimate layer. An approximation to the euclidean distance to the decision boundary in the input space for a multi-class classification problem can be computed as

$$\frac{|f_i(x) - f_j(x)|}{||\nabla_x f_i(x) - \nabla_x f_j(x)||_2}$$

where $f(x) = \sigma^{-1}C_i(y_i|x_i, s_i))$ (Elsayed et al., 2018). This implicitly assumes a decision boundary at 0.5 and needs to be redefined slightly for binary classification. For this master's thesis, approximate the euclidean distance of a point x_i to the decision boundary as

$$\frac{|f(x)|}{||\nabla_x f(x)||_2}$$

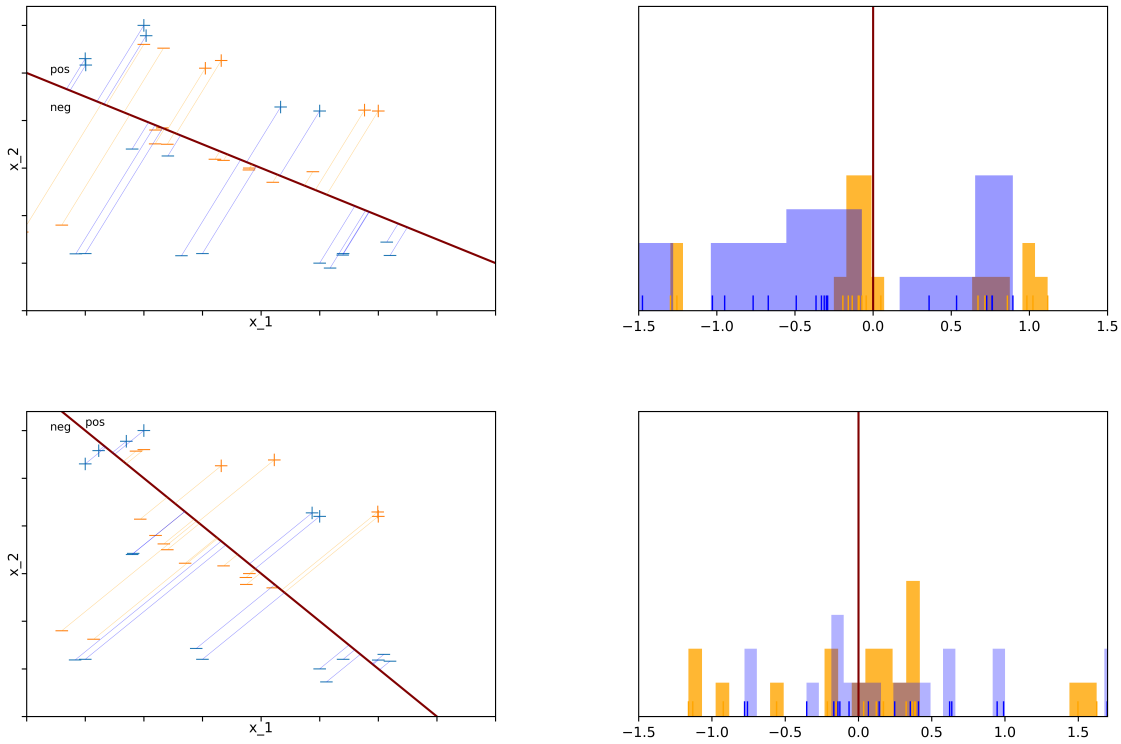

Figure 3: Joint distribution of margins and sensitive variable

Orange and blue are sensitive groups, (+,-) mark positive and negative labels. Neg and pos are negative and positive sides of decision boundary. The first decision boundary induces very different margin distributions for orange and blue. Many orange samples are concentrated just on the negative side of the decision boundary. The second decision boundary induces more equal margin distributions. Sensitive group membership of points near the decision boundary is not as imbalanced.

For a thresholded classifier, this master's thesis proposes approximating the euclidean distance of a point x_i to the decision boundary as

$$\frac{|\sigma^{-1}(t) - f(x)|}{||\nabla_x f(x)||_2}$$

where $f(x)$ is defined as $f(x) = \sigma^{-1}C(y_i|x_i, s_i))$ with σ^{-1} the sigmoid function and t the threshold for the sensitive group s_i to which the point x_i belongs, see the appendix for a derivation.

5 Evaluating Performance and Generalization

Performance of the algorithms is evaluated on five data sets. The sensitive variable for the adult data set [5] is gender, the dependent variable is binarized income, with the threshold set at 50000 $. Additionaly, a downsampled version of the adult data set with higher positive rates was created in order to better differentiate between the effect of sample size and data set characteristics on performance and discourage degenerate classifiers. For the Credit data set [6] credit default is the dependent variable. The sensitive variable is a binary age

[5] available online via https://archive.ics.uci.edu/ml/datasets/adult
[6] available online via https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

variable with the threshold at 26. With only 999 data points it allows gauging perfmance in the small data regime. The reincidencia (Tolan et al., 2019) data set contains information on recidivism of catalan youth, the sensitive variable is gender. For the compas data set (ProPublica, 2017) the dependent variable is again recidivism and the sensitive variable is white/non-white and. All variables are normalized to fall within the $(0,1)$ range in order to increase performance of gradient descent. Categorical variables were encoded as dummies.

	Credit	Reincidencia	Compas	Adult downsampled	Adult
n	999	4753	6172	6000	45222
n_{s_0}	149	3908	4069	1571	14695
n_{s_1}	850	845	2103	4429	13527
$features$	20	17	17	12	12
$Baseline$	0.7	0.65	0.54	0.5	0.75
$pos.\ rate_{s_0}$	0.41	0.63	0.49	0.27	0.11
$pos.\ rate_{s_1}$	0.28	0.8	0.39	0.58	0.31

Table 1: Overview of the data sets

Baseline refers to the accuracy of a degenerate classifier, i.e. a classifier that classifies all points as one class

Some of the algorithms evaluated need extensive hyperparameter optimization. For this, an objective needs to be set up to determine the trade-off between accuracy and fairness an algorithm is to be optimized for. Here this master's thesis proceeds similarly to Edwards and Storkey (2015). For every data split, 100 classifiers are trained on a training set, performing random search for hyperparameter optimization. For thresholding the best most accurate model should be selected and then it's thresholds adjusted. If fair classification was as simple as adjusting threshold, this should give the best results. Thus, for the thresholded neural network, 25 classifiers were trained on training data and one classifier was selected on validation data based on it's accuracy. Then, thresholding was performed 100 times on this one classifier with increasing values of p up to 0.99. For the logistic regression only one classifier needed to be trained, due to the convexity of it's loss function. Here as well, thresholds were adjusted for 100 different values of p. Performance is computed as

$$perf = \sum_i (\hat{y}_i = y_i) - t * (1 - di), \;\; t \in [0,3]$$

for 100 equally spaced values of t in the interval, where di is the classifier's disparate impact [7].

Then, without retraining, the performance of the best-performing model on the test set for a particular t is computed [8]. To reduce the variance of the estimated performance, the

[7] Edwards and Storkey (2015) instead compute the trade-off between accuracy and Calders-Verwer metric (Calders and Verwer, 2010), i.e. the difference between positive rate for the sensitive groups

[8] Models selected on training data and then evaluated on test data did not exhibit worse performance, except for the

procedure is repeated 10 times over for the Credit data set and 5 times over for all data sets except the large Adult data set, for which it was repeated 3 times. Generally, a smaller test set leads to higher variance of the performance and repeating testing on different splits lowers variance of the perfomance estimate Rodriguez et al. (2009). Therefore, the training set size was chosen as sixty percent of the size of the data, test and validation as twenty percent. Classifier with an average of sensitive group wise positive rates smaller than 0.05 and larger 0.95 were omitted in order to avoid quasi degenerate classifiers, i.e. classifiers that only predict one label.

5.1 Fair performance

This section section gives an overview of the most important empirical results. First results for the linear models are reported, then the results for the non-linear models. Only a subset of the results are depicted in the main text, the remaining results can be found in the appendix. Plots of test performance show the average test performance over all folds computed as described in the previous section. All other quantities (training performance, fairness, etc.) show the average over data splits over models selected on validation data and then tested on test data based on their performance. In other words, training performance for the models for which test performance was reported. Disparate impact is reported as $di_{train} - di_{test}$, the difference between training and test performance.

With the exception of the compas data set, the thresholded logistic regression dominates the in-processing logistic regressions on training data for all fairness-accuracy trade-offs. Yet on test data, the tresholded classifier is outperformed by the adversarially constrained logistic regression on all data sets at higher values of the trade-off parameter t except on the large adult data set. The drop in performance on test data of the thresholded classifier at higher values of t can be explained by it's drop in fairness. Fairness of the thresholded logistic regression also drops on the downsampled adult data set. No model achieves good performance on the credit data set. Even for small values of t, performance drops below the baseline. On both the downsampled adult data set and the large adult data set, the constraints proposed by Kamishima et al. (2012) and Zafar et al. (2017b) can barely improve fairness. Yet on the compas data set, they do not only show good performance, but also generalize well. The same holds true for the credit data set, but to a much smaller degree. While the adversarially constrained logistic regression achieves decent fairness on all data sets and generalizes well, it does so at the cost of achieving low training performance. For some data sets, the best models on validation data barely achieve higher performance than the baseline, i.e. the accuracy of a degenerate classifier.

For neural networks, the post-processing by thresholding was tested against the variational fair autoencoder and the adversarially constrained model. On the credit data set, the adversarially constrained model and the variational fair autoencoder did almost always col-

adversarially constrained models and the variational fair autoencoder, which showed marginally worse results.

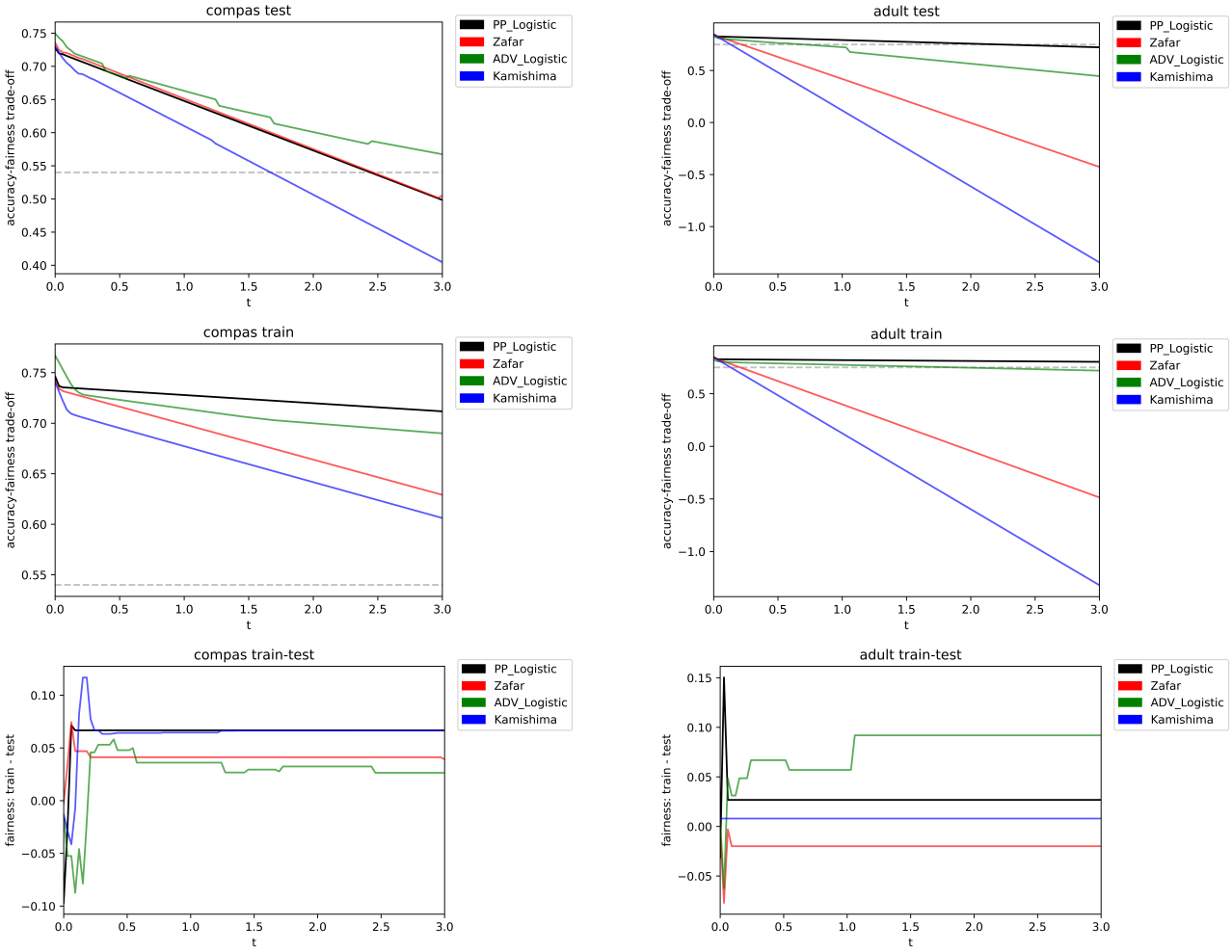

Figure 4: Performance of linear models

Plots show from top to bottom test performance, train performance and the difference between disparate impact on the training set and disparate impact on the test set. Test performance is reported as the average over all folds and accuracy-fairness trade-offs. Data sets are compas in the first column and adult in the second column. Dotted line is the baseline. Post-processing achieves higher test performance than in-processing on the adult data set, but lower test performance on the compas data set. This is due to the drop in accuracy from test to train data in the smaller data set. Adult is the only data set where performance of the adversarial model drops more.

lapse to a degenerate or quasi-degenerate classifier. For the other data sets, the picture is similar to the linear case. On training data, thresholded neural networks outperform the other classifiers on all training data sets. Compared to the linear case, the difference in performance on training data is reduced, however. This is due to the variational fair autoencoder and the adversarially constrained network not only achieving high levels of fairness, but also at a good trade-off against accuracy. On test data, the thresholded neural networks are outperformed on all data sets for higher values of t except for the downsampled adult data set, where the adversarial model did not achieve good levels of fairness. This can again be explained by the gap in fairness between train and test data. On all data sets except the downsampled adult data set, impact disparity is low for the adversarially constrained neural network and the variational fair autoencoder. Compared to linear models, both the variational fair autoencoder and the adversarially constrained model achieve a somewhat better fairness-accuracy trade-off and more reliably reduce bias. This holds true even when com-

paring against the adversarially constrained logistic regression, which also reduces impact disparities by reducing dependency between predictions and the sensitive variable.

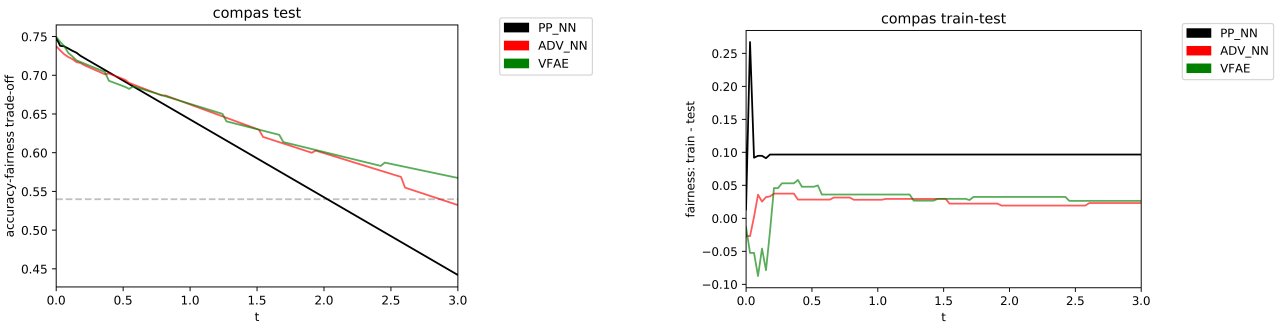

Figure 5: Performance of non-linear models

Plots show test performance on the left and the difference between disparate impact on the train set and on the test set on the right. Test performance is reported as the average over all folds and accuracy-fairness trade-offs. Post-processing is again outperformed on test data due to the differnce between train and test performance.

From the results, it is not clear to what extent performance depends on sample size. Performance is abysmal on the credit data set, with a test size of just 200 observations. Overall performance on the reincidencia data set is worse than on the larger data sets, with the exception of the Kamishima et al. (2012) and Zafar et al. (2017b) models. While these models perform well on the compas data set, they barely improve performance on the adult data sets. Performance disparities between the downsampled adult data set and the adult data set are suprising . On the downsampled adult data set, the thresholded neural network attains the highest test performance, yet on the full sample, it does not. Models trained with the constraint proposed by Zafar et al. (2017b) and Kamishima et al. (2012) show equally bad performance on both data set.

During training, it was noted that the adversarially constrained models and the variational fair autoencoder often collapsed to degenerate classifiers. While this is probably unavoidable to some extent, the number of models collapsing was extreme, with the minimum of uncollapsed models only at 50 models for the adversarially constrained logistic regression on the compas data set. While the constraint proposed by Kamishima et al. (2012) is non-convex, the classifiers did still reliably converge to similar minimums. For the thresholded classifier, it was noted during training that results on validation data did not seem to improve much for the last increases of p. This points towards minor adjustments of the decision boundary on training data not generalizing to unseen data. While the thresholded neural network performs better than the thresholded logistic regression, it does not perform better relative to it's comparison group. This points towards the slightly improved estimation of $C(y|x,s)$ not strongly improving thresholding performance.

5.2 Generalization and mutual information

From the results it is clear that fairness of the adversarially constrained models and the variational autoencoder generalizes better than the fairness of the thresholded classifier.

Whenever they achieve good performance on training data, fairness of the constrained logistic regression proposed by Zafar et al. (2017b) also generalizes better, but the difference is of smaller magnitude. In section 4 it was hypothesized that a lower mutual information between the margin distribution and the sensitive variable could lead to better fairness generalization. This cannot be confirmed by the results.

In order to test the hypothesis, dependence between the margins and s needs to be quantified. Dependence can be quantified via mutual information

$$I(m(x,s),s) = D_{KL}(P(m(x,s),s)||P(m(x,s)) \otimes P(s))$$

Mutual information can be estimated from samples based on entropy estimates from k-nearest neighbor distances (Kraskov et al., 2004). In addition to mutual information and to rule out any misbehaviour of the estimator, a second measure of dependence was computed. Gradient boosted trees were trained to predict the sensitive variable from the margins for a subset of the models. Then, the error was computed. A high error implies low dependence. As very complex tree models had to be trained in order to learn the non-linear dependencies, the procedure was very computationally intense. It was therefore only implemented in order to verify the strength of dependency as quantified by the mutual information estimate. Results on that subset largely agreed with the mutual information estimate.

For all models selected on validation data, the mutual information between the margins and the sensitive variable was computed. For every data split, only a small number of models are optimal for one of the accuracy-fairness trade-offs. The number of selected models from one split is mostly below five. This can be explained by only a few models achieving high fairness. Yet this also entails that comparing characteristics of these models is inherently noisy. Nonetheless, it will be investigated how these models compare with respect to the joint distribution of their margins and the sensitive variable in order to test the hypothesis presented in section 4.

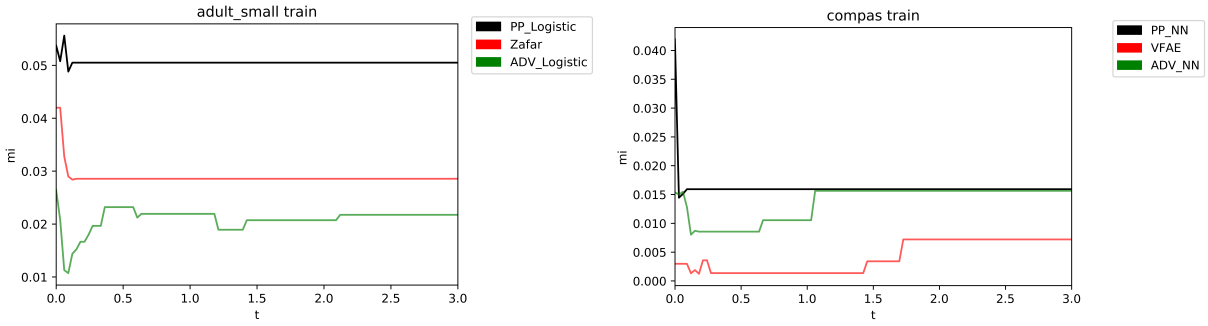

Figure 6: Average mutual information for models selected on validation data

Plots show the average mutual information for models selected on training data. Data sets are the downsampled adult data set with linear models on the left and compas with non-linear models on the right. Corresponding figures for the remaining data sets can be found in the appendix.

Contrary to expectation, the margin distribution of the variational fair autoencoder and

of the adversarially constrained neural network is not any less dependent on s than the margin distribution of the post-processed neural network. Overall, the ranking of mutual information shows great variation across data sets and within data set. While for the reincidencia data set the estimate of mutual information is lower at most thresholds, for the adversarially constrained neural network it is higher on most thresholds on the adult data set. Yet on the adult data set, generalization performance is better. On the downsampled adult data set, for the adversarially constrained models selected on validation data for a given accuracy-fairness trade-off, mutual information even increases with fairness. Magnitude of mutual dependence varies quite strongly between data sets, ranging from 0 on the compas data set for adversarially constrained models to 0.14 on the adult data set for the adversarially constrained neural network. This could of course be due to the noise of the small sample size, but clearly results for neural networks do not support the hypothesis. For linear models, mutual information is highest for the thresholded classifier for all data sets but some values of the trade-off t on the credit data set. Mutual information could not be computed for the models with the Kamishima et al. (2012) constraint, as the AIF360 implementation did not give access to the predicted logits. The author did decide not to reimplement the model as results and constraints were comparable to the Zafar et al. (2017b) model.

In order to examine if the magnitude of the margins leads to different generalization performance, the average absolute value of the margin of the 5 percent of points closest to the decision boundary was computed for all selected models. No clear trend could be observed. On the compas data set, the variational fair autoencoder's margins are of higher magnitude. Yet on e.g. the adult data set, the thresholded neural network places the decision boundary further away from the closest points. As the number of models selected on validation data is small, this cannot be interpreted as disproving an effect of the margin's magnitude on generalization performance, but does suggest no strong relationship.

In order to further test for a link between the margin distribution, s and generalization error, the joint distribution of di_{train}, s and $di_{train} - di_{test}$ was examined for the adversarially constrained models and the variational fair autoencoder based models. While obviously it would be best to take into account the thresholded models, this was not possible due to their high similarity. Models were extremely similar as thresholding only slightly shifts the decision boundary. Retraining different models was not a solution as the neural networks tended to converge to similar minimums and the logistic regression obviously always converges to the same minimum. The highly non-linear variation fair autoencoder and adversarially constrained models end up in very different minima, however. If the hypothesized link between generalization error and the joint distribution of the margins and s exists, it should show for these models. Disparate impact on the training set was taken into account as it might have a negative effect on generalization, as higher values of fairness could generalize worse. It

could therefore obfuscate a link between mutual information of s and the margins on the one hand and the generalization error on the other hand. Figure 7 shows that for all non-quasi-degenerate models on the compas and small adult data set, there exists no such link. Generalization error is evenly spread along the range of estimated mutual information. No clear pattern is visible. Additonally , generalization error was regressed on mutual information and the magnitude of the margins of the 5 percent of points closest to the decision boundary. Less than two percent of the variance could be explained. This held true when filtering by model types.

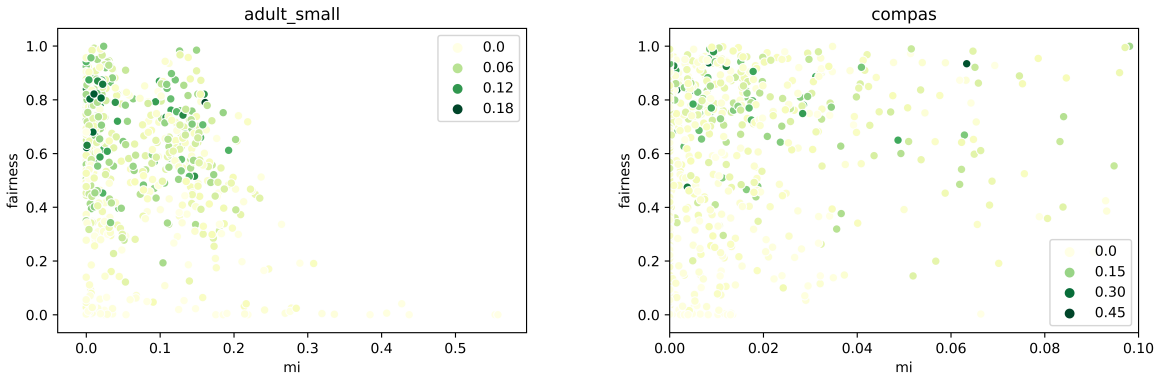

Figure 7: Generalization error and mutual information

All non-quasi-degenerate adversarially constrained and variational fair autoencoder models trained on all splits for small adult (left) and compas (right) data sets. The scatterplot shows the joint distribution of di_{train} and $I(m(x,s),s)$. **Hue is the generalization error** $di_{train} - di_{test}$. No patterns are discernible.

How then could the better generalization performance of the adversarially constrained models and the variational fair autoencoder be explained? Considering the large number of models trained for each data split, it could be that due to their high non-linearity, the adversarially constrained models and variation fair autoencoders have very diverse predictions. Assuming that the covariate shift between test data and training data goes in the same direction as the covariate shift between validation data and training data, it could be that the pool of models is large and diverse enough to always allow choosing one model that by chance generalizes well.

6 Conclusions

This master's thesis compared fair thresholded classifiers with fair in-processing classifiers. Drawing on previous research questioning the necessity of algorithms that specifically optimize for fairness, a thresholding as an integer linear program was proposed. It's performance dominates all in-processing classifiers on all training data sets. Yet on most data sets, in-processing classifiers based on reducing statistical dependence between predictions and features outperformed the thresholded classifiers by some margin. This was true both in the linear case, where the margin classifier was a logistic regression, as in the case of a neural network. It was shown that this is mainly due to in-processing classifier's fairness generalizing better to unseen data.

A link was made to existing literature on generalization and properties of the margin distribution. It was argued that if a classifier's margin distribution is independent of the sensitive variable, it should be robust to noisy resampling of points near the decision boundary and therefore generalize better. Motivated by this observations, mutual information between the sensistive variable and the margin distribution was computed for all models. While the on validation data selected adversarial logistic regressions did have lower mutual information than the thresholded models on all data sets, the comparison to the model proposed by Zafar et al. (2017b) was not as clear. In the case of neural networks, no clear trend did emerge. By analyzing the joint distribution of fairness, no link between mutual information and the joint distribution of the margins and s could be found. Nor could any of the variation in generalization performance be linearly explained by the the magnitude of the margins and mutual information by a regression.

One important implication of these findings is that achieving fairness on training data is not sufficient for achieving fairness on test data. The results point towards a simple constraint that merely imposes equality in positive rates being insufficient to achieve fairness that generalizes. It cannot be ruled out, however, that the highly non-linear variational fair autoencoder and adversarially constrained models simply gain an advantange by having a more diverse pool of models to select from. As such, no clear cause of the generalization gap could be isolated. Yet the empirical results of the thesis clearly show that such a gap exists. This gap even has practical consequences. If legal requirements only allow impact disparities up to some magnitude p, a classifier must be p-fair on unseen data and not only on training data. Studying the generalization gap in fairness could help gauging the fairness loss incurred on new data. This is of special importance in the real world where the data the algorithm is applied on might not be drawn from the same distribution as training data, due to e.g. the distribution changing over time. Then, even when working with large data, thresholding might not be a viable route to take as it's poor generalization properties might also make it susceptible to covariate shift.

The results imply that there are lower limits to the size of data if high levels of fairness are required. This has practical consequences, as often in the real world training data can be of limited size due to the high cost of data collection and labeling. Unless fair algorithms are developed that achieve higher levels of fairness on small data, collecting additional data would be inevitable to meet fairness requirements.

While the thesis did not find a clear link between mutual information of the margin distribution and s on the one hand and generalization error on the other hand, the results do not suffice to disprove such a link. There are two imprecisions in the analysis. For one, estimation of mutual information is notoriously difficult and can be unreliable. Secondly,

distance to the decision boundary in the input space for neural networks was approximated by linearizing the transformation at a given point. Linearization could potentially give very imprecise estimates of distance for points far away from the decision boundary. This could lead to existing trends in the margin distribution being obfuscated by the linearization.

During training, it was noted that adversarially constrained models and the variational fair autoencoder frequently collapse to quasi-degenerate models, i.e. models that classify all or almost all points as positive or negative. This could probably to some extent be mitigated by choosing a narrower searchspace. That would however require even closer monitoring of validation performance during training, which might not be feasible when training several models on different data sets. It might be possible to reduce the frequency of collapse by some form of label smoothing, something that was not investigated in this thesis.

The results do not dispel the idea of using post-processing as a means to achieving fairness that generalizes. While a simple thresholding shows poor generalization results, enforcing a more robust fairness might also be achievable by post-processing. This seems fruitful from a performance point of view as post-processing always achieves better performance on training data. This points towards the algorithm not necessarily getting stuck in minima suboptimal for fairness. Such post-processing would however most likely require transformations of the predictions and not just shifts of the thresholds, which does not generalize well.

References

Adel, T., Valera, I., Ghahramani, Z., and Weller, A. (2019). One-network adversarial fairness. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. (2018). A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*.

Altman, A. (2016). Discrimination. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition.

Barocas, S., Hardt, M., and Narayanan, A. (2017). Fairness in machine learning. *NIPS Tutorial*.

Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al. (2018). Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.

Beutel, A., Chen, J., Zhao, Z., and Chi, E. H. (2017). Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*.

Botros, P. and Tomczak, J. M. (2018). Hierarchical vampprior variational fair auto-encoder. *arXiv preprint arXiv:1806.09918*.

Calders, T. and Verwer, S. (2010). Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292.

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*, pages 3992–4001.

Celis, L. E., Huang, L., Keswani, V., and Vishnoi, N. K. (2019). Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 319–328. ACM.

Cotter, A., Gupta, M., Jiang, H., Srebro, N., Sridharan, K., Wang, S., Woodworth, B., and You, S. (2018). Training fairness-constrained classifiers to generalize.

Dale, S. B. and Krueger, A. B. (2014). Estimating the effects of college characteristics over the career using administrative earnings data. *Journal of human resources*, 49(2):323–358.

Dastin, J. (2018). Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM.

Dwork, C., Immorlica, N., Kalai, A. T., and Leiserson, M. (2018). Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133.

Edwards, H. and Storkey, A. (2015). Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*.

Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. (2018). Large margin deep networks for classification. In *Advances in neural information processing systems*, pages 842–852.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 329–338. ACM.

Ganin, Y. and Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520.

Hardt, M., Price, E., Srebro, N., et al. (2016). Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Huang, L. and Vishnoi, N. K. (2019). Stable and fair classification. *arXiv preprint arXiv:1902.07823*.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. (2018). Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*.

Kamiran, F. and Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kleinberg, J., Ludwig, J., Mullainathan, S., and Rambachan, A. (2018). Algorithmic fairness. *AEA Papers and Proceedings*, 108:22–27.

Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.

Lipton, Z., McAuley, J., and Chouldechova, A. (2018). Does mitigating ml's impact disparity require treatment disparity? In *Advances in Neural Information Processing Systems*, pages 8125–8135.

Liu, L. T., Dean, S., Rolf, E., Simchowitz, M., and Hardt, M. (2018). Delayed impact of fair machine learning. *arXiv preprint arXiv:1803.04383*.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. (2015). The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. (2018). Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*.

Melguizo, T. (2008). Quality matters: Assessing the impact of attending more selective institutions on college completion rates of minorities. *Research in Higher Education*, 49(3):214–236.

ProPublica (2017). Compas recidivism risk score data and analysis.

Raff, E. and Sylvester, J. (2018). Gradient reversal against discrimination. *arXiv preprint arXiv:1807.00392*.

Rodriguez, J. D., Perez, A., and Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575.

Sattigeri, P., Hoffman, S. C., Chenthamarakshan, V., and Varshney, K. R. (2018). Fairness gan. *arXiv preprint arXiv:1805.09910*.

Tolan, S., Miron, M., Gómez, E., and Castillo, C. (2019). Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Xu, D., Yuan, S., Zhang, L., and Wu, X. (2018). Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE.

Zafar, M. B., Valera, I., Rodriguez, M., Gummadi, K., and Weller, A. (2017a). From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*, pages 229–239.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. (2017b). Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970.

Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM.

Appendices

A Notation and definitions

Symbol / Word	Definition
performance	the fairness-accuracy trade-off
$y \in \{-1, 1\}$	the labels
X	the matrix of features
$s \in \{0, 1\}$	the sensitive variable
$m(x, s) \in \mathbb{R}$	the margin (signed perpendicular distance to decision boundary)
$C(y\|X, s) \in (0, 1)$	The conditional density estimated by a classifier
$\hat{y} \in \{0, 1\}$	the classifier's predictions, obtained by thresholding $C(Y\|X, s)$
$di(\hat{y})$	The disparate impact of a classifier's output
p	The amount of disparate impact allowed
$P_{sample}(z)$	The empirical distribution of z
$\sigma(z)$	$\frac{1}{1+e^z}$

B Diagram VFAE training process

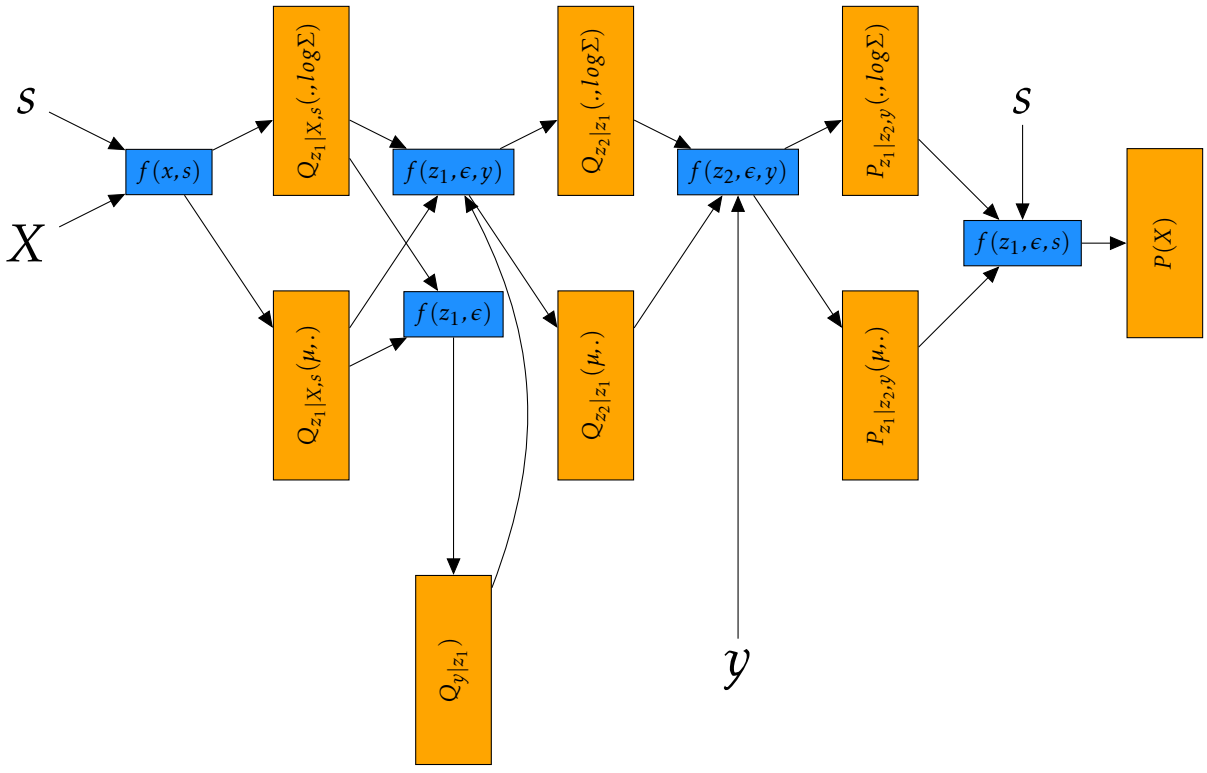

Figure 8: A diagramm of the variational fair autoencoder. Orange are estimated variables/ parameters. Blue are neural networks. The ϵ denotes a draw from a standard normal distribution (once per layer and sample).

C Post-processing comparison to Lipton et al. (2018)

Lipton et al. (2018) describe a simpler thresholding algorithm than the author for satisfying the p-rule. Here, to all $c(y|x,s=0) > 0.5, (c(y|x,s=1) < 0.5$ with a higher predicted positive rate for the $s=0$ group, Lipton et al. (2018) assign a score that weights data points by the model's certainty and by the group's size. Scores are $\frac{p}{n_{s_0}(2 * c(y|x,s=0)-1)}$ and $\frac{1}{n_{s_1}(1-2 * c(y|x,s=1))}$. Then, predictions for the data points with the highest score are flipped, until the p-rule is satisfied. This algorithm does not take into account the true labels. When the model comits systematic error, e.g. by having different accuracy for different groups or by having low accuracy in certain regions, this can lead to suboptimal thresholds.

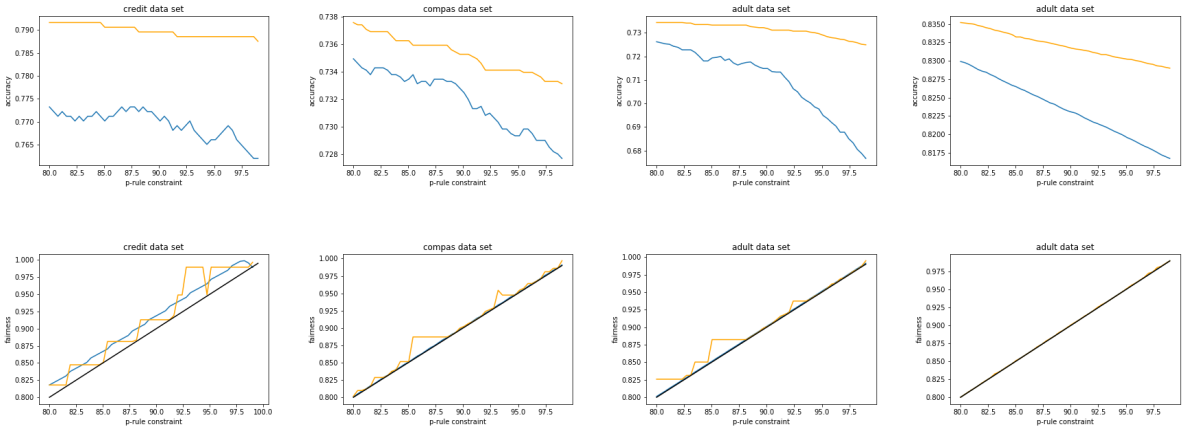

Figure 9: Performance comparison Lipton et al. (2018) post-processing and the author's implementation on top of a logistic regression on training data

Orange is the author's ilp thresholding, blue Lipton et al. (2018), black (only for the fairness plot) the fairness constraint. The author's thresholding achieves better performance on all training sets, albeit with a small to miniscule margin.

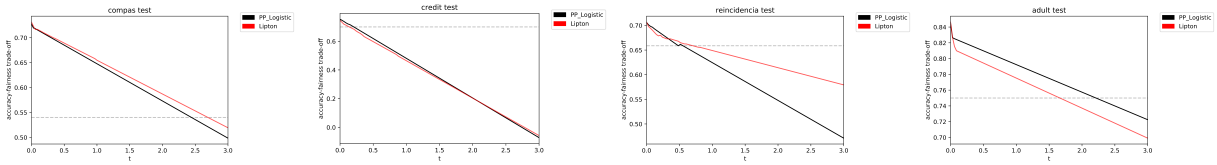

Figure 10: Performance comparison Lipton et al. (2018) post-processing and the author's implementation on top of a logistic regression on test data

Black is the author's ilp, red Lipton et al. (2018). Lipton et al. (2018) generalizes equally well to unseen data. Performance is better for the reincidencia data set and worse for the adult data set. Performance on compas and credit is equal. Performance disparities are not nearly as large as they are to the in-processing classifiers and their is no thresholding that dominates the other, as in-processing classifiers that reduce dependence between s and the margins do.

D Suboptimal decision boundary

This section gives a brief example of a data set which induces a linear decision boundary that optimizes accuracy but does not optimize the accuracy fairness trade-off over the set of linear classifiers. By thresholding, the decision boundary can only be shifted parallelly. The optimal decision boundary maximizing accuracy under fairness constraints must not be parallel, however. Take the set of 12 points in figure 11. The Logistic regression classifier discriminates between positive and negative labels based on 3 variables, x_1, x_2, s, where s is the sensitive variable. As s is binary, it's effect on the decision boundary in the x_1, x_2 space is a parallel shift. Blue and orange are the decision boundaries drawn by the logistic regression in the x_1, x_2 space. The decision boundary achieves optimal accuracy for a linear classifier, correctly classifying all but one of the data points. The best parallel shift of this decision boundary can only achieve 9 correctly classified points (out of 12) under the constraint of equal positive rates. The green decision boundary would however yield a fair linear classifier with 10 correctly classified points. Thus the classifier gets trapped by maximizing for accuracy first, which could be prevented by joint optimization for accuracy and fairness.

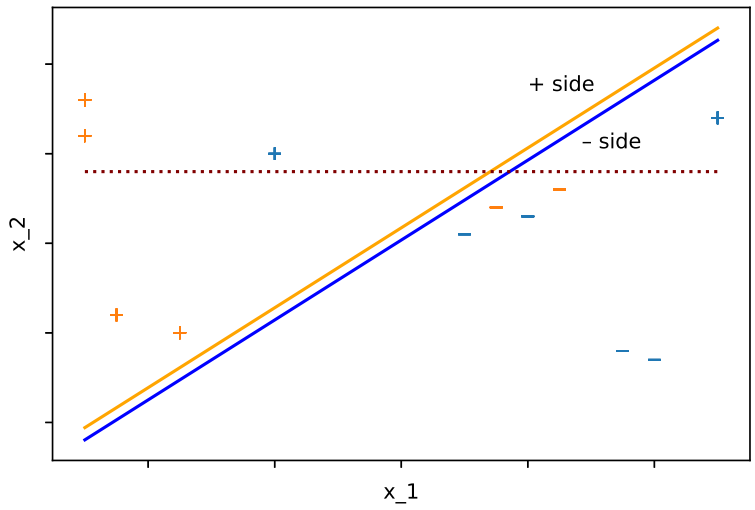

Figure 11: Post-processing stuck

textbfOrange and blue are sensitive groups, (+,-) mark positive and negative labels. In this figure, the original decision boundary is suboptimal for a fair linear classifier. Blue and orange are decision boundaries of a logistic regression for sensitive group blue and sensitive group orange. Green would be an optimal decision boundary for a fair classifier

E Reproduction results

E.1 Edwards and Storkey (2015)

Presented in figure 12 is a comparison of the author's pytorch implementation of Edwards and Storkey (2015) and the results reported in their paper. Compared here are results for the adult data set. Edwards and Storkey (2015) train 100 models on the train set, compute the performance as $\sum_i \hat{y}_i = y_i - t * |\sum_{i:s_0} c(x_i)/n_{s_0} - \sum_{i:s_1} c(x_i)/n_{s_1}|$ for $t \in [0,3]$ for varying values of t on the validation set and then compute the performance on the test set (without retraining). The exact same approach was taken by the author. Reproduction results cannot expected to be exact, but the pytorch implementation achieves an equal level of performance. The trade-off between accuracy and fairness is somewhat different, which would be expected given that the distribution over hyperparameters is uniform and has a large support. While models selected in the (Edwards and Storkey, 2015) paper achieve higher accuracy, model trained for the thesis achieve higher fairness. Due to the high non-linearity of the cost function, some differences are expected. Note that fairness here refers to the difference between positive rates, compare (Calders and Verwer, 2010).

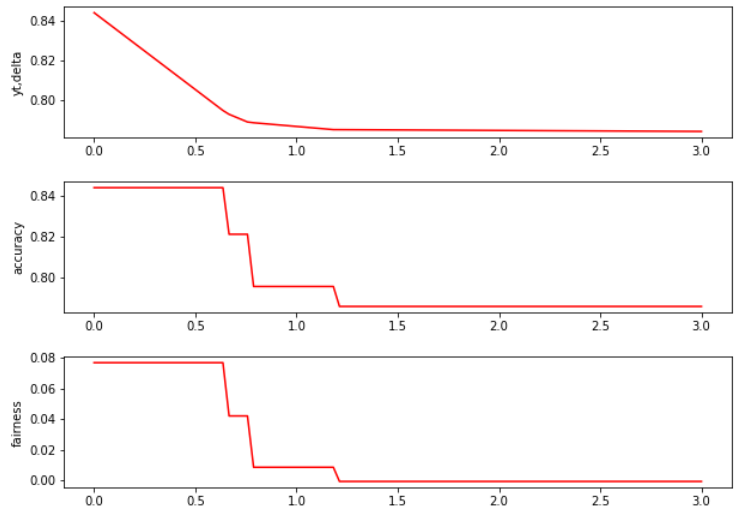

Figure 12: The same performance measures as reported by edwards for the author's pytorch implementation.

E.2 Louizos et al. (2015)

In their paper, Louizos et al. (2015) do report results on a test set, after models were trained on a training set and selected (but not retrained) on a validation set. They do not describe how performance on the validation set was computed, merely stating that the primary aim was fairness, but predictions should not be degenerate. This is a problem for replication, as such a criterion of performance is necessary for model selection. Instead of selecting models on validation data, for this reproduction 20 models are trained on a train set and their performance on the validation related to the test set. If good models that would be selected on validation data achieve comparable performance as the performance reported by Louizos et al. (2015) on test data, then their findings could be replicated.

The models are trained for different data set splits and results are reported on the validation set and test set. To keep results comparable to Louizos et al. (2015), train, validation and test data are of the same size as in their paper. Data preparation is the same as in the paper. Results shown in figure 13 are comparable to what is reported by Louizos et al. (2015). Note that fairness is defined by the absolute difference between positive rates (Calders and Verwer, 2010), not disparate impact.

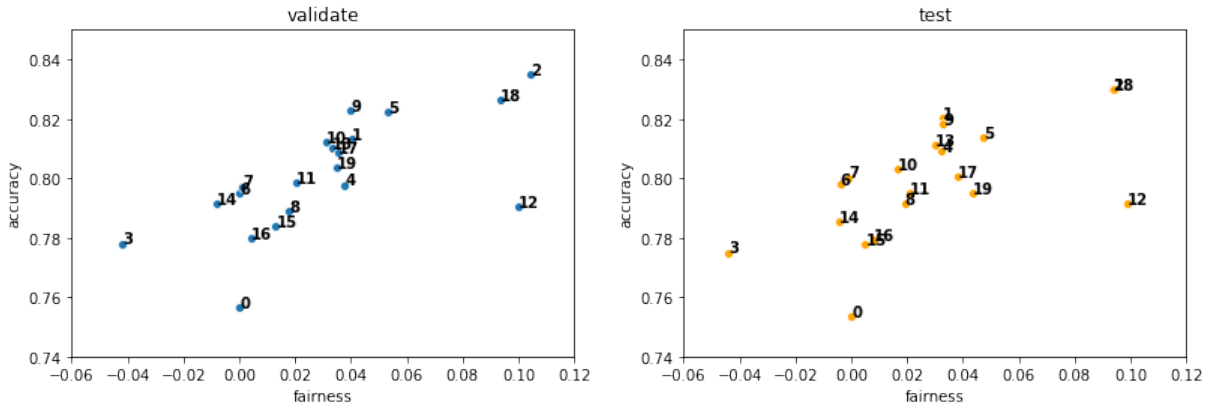

Figure 13: Resuls on validation and test data

A model chosen on validation data with a comparable fairness-accuracy trade-off as Louizos et al. (2015) report in their paper on the test set also achieves similar performance on the test set.

F Distance to the decision boundary of a thresholded neural network

Elsayed et al. (2018) define the decision boundary as $db = \{x | f_i(x) = f_j(x)\}$ where f_i gives the score for class i Then, for any point $x \in \mathbb{R}^p$, it's euclidean distance to the decision boundary is defined as:

$$d = min_\delta ||\delta||_2 \; s.t. \; f_i(x+\delta) = f_j(x+\delta)$$

with $\delta \in \mathbb{R}^p$. By linearizing at $\delta = 0$ one gets

$$min_\delta ||\delta||_2 \; s.t. \; (\nabla_x f_i(x) - \nabla_x f_j(x))'\delta = f_j(x) - f_i(x)$$

They can prove that [9] the solution to any optimization problem of the form $min_\delta ||\delta||_2 \; s.t. \; b'\delta = a$ is $\frac{|a|}{||b||_2}$. Therefore, the distance to the decision boundary of a neural network in the input space can be approximated as (Elsayed et al., 2018):

$$\frac{|f_i(x) - f_j(x)|}{||\nabla_x f_i(x) - \nabla_x f_j(x)||_2}$$

For this master's thesis, this first needs to be defined for the two-class case and then for the case where the threshold is not at 0.5. For the two-class case, define the decision boundary as $db = \{x | f(x) = \sigma^{-1}(0.5)\}$, where σ is the sigmoid function. Then the optimization problem becomes:

$$min_\delta \|||\delta||_2 \; s.t. \; f(x+\delta) = 0$$

and the distance to the decision boundary is:

$$\frac{|f(x)|}{||\nabla_x f(x)||_2}$$

Equivalently, when the threshold is at t, define the decision boundary as $db = \{x | f(x) = \sigma^{-1}(t)\}$, the optimization problem becomes:

$$min_\delta \|||\delta||_2 \; s.t. \; f(x+\delta) = \sigma^{-1}(t)$$

and the distance to the decision boundary is:

$$\frac{|\sigma^{-1}(t) - f(x)|}{||\nabla_x f(x)||_2}$$

[9] see supplementary material of the paper

G Results

G.1 Performance

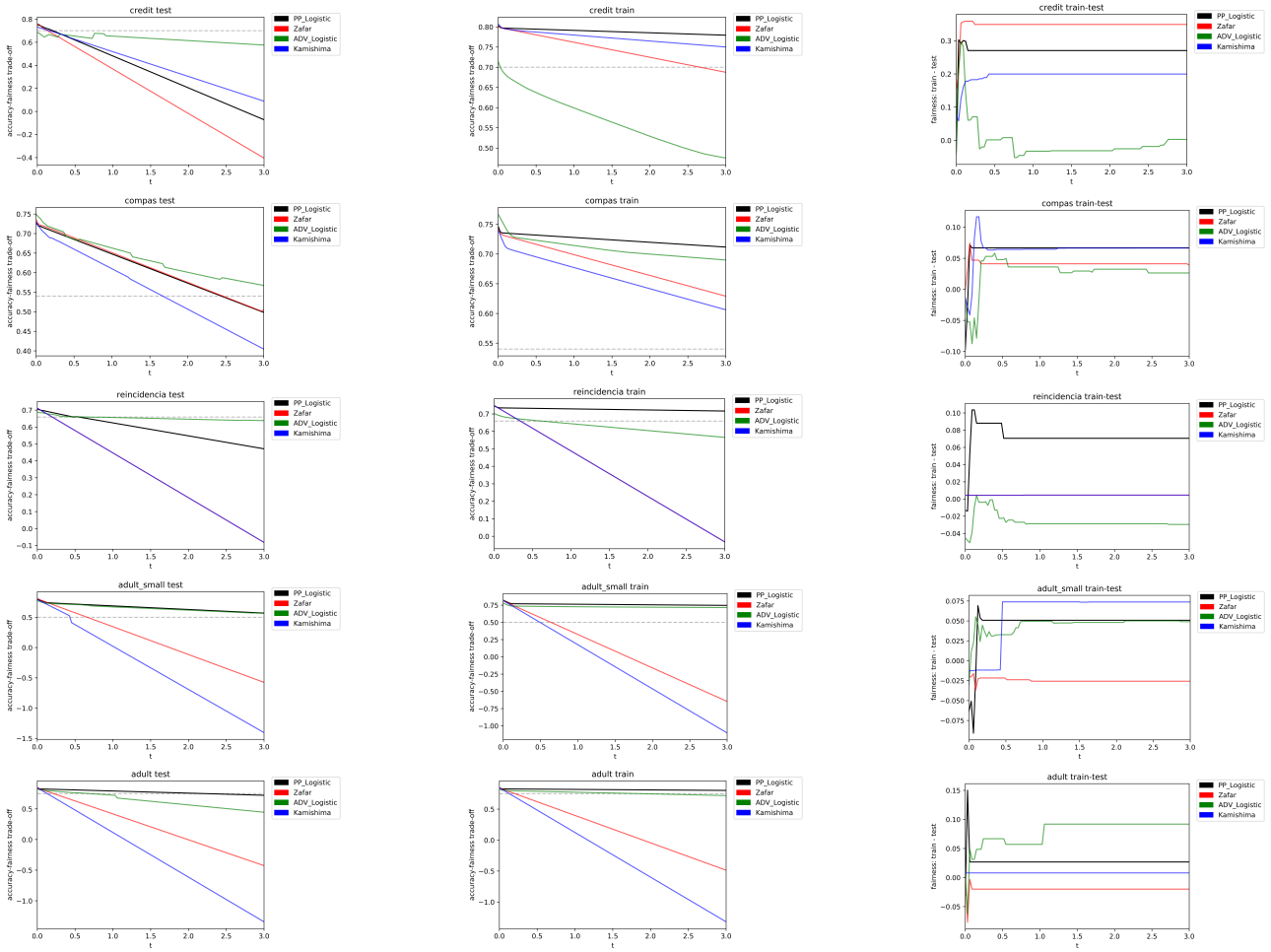

Figure 14: Performance and fairness of linear models

Columns left to right: Test performance, train performance and difference between fairness on training data and fairness on test data The thresholded logistic regression dominates on training data but is surpassed by the adversarial model on test data for higher values of t. This is due to a smaller generalization gap.

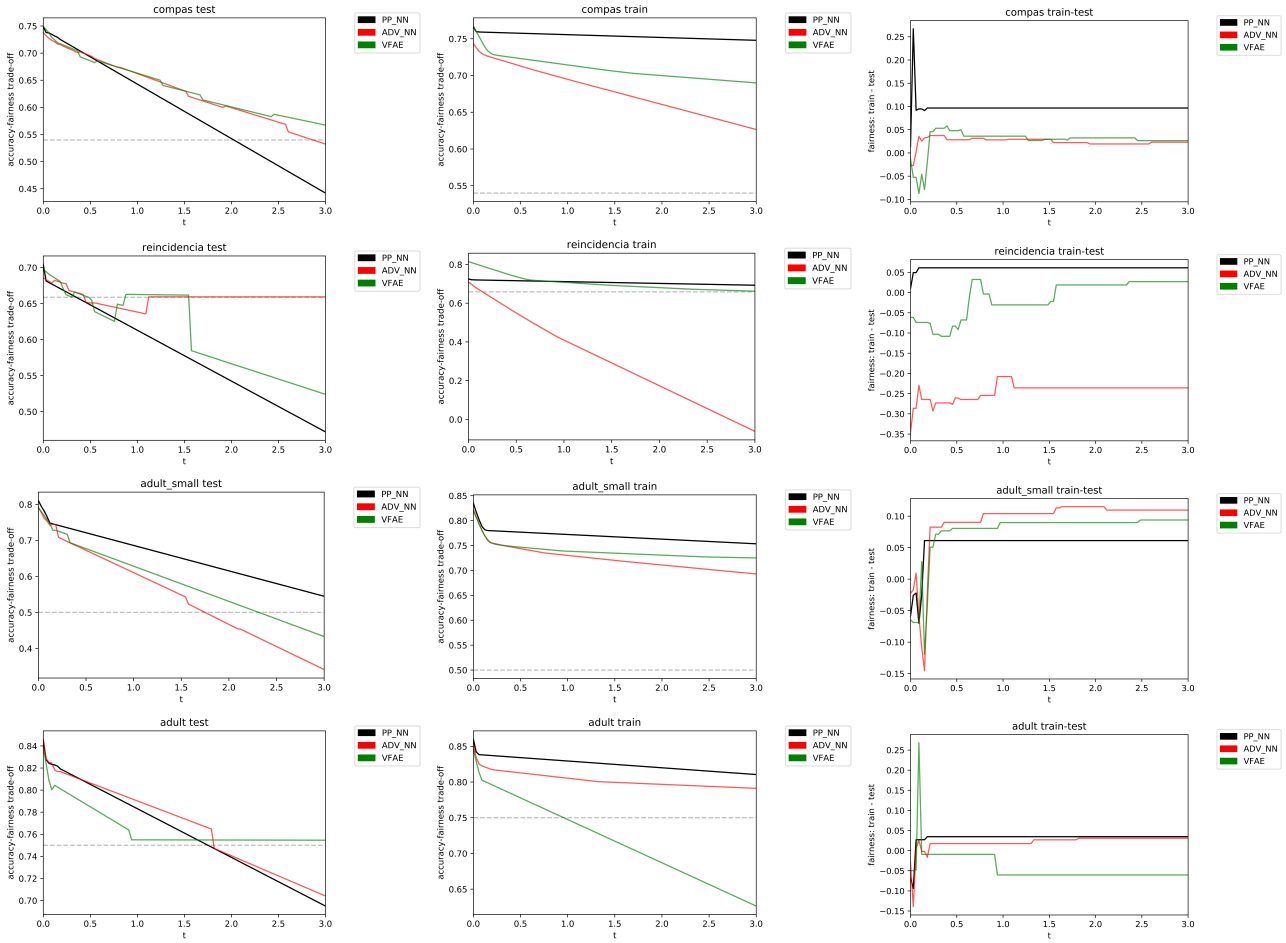

Figure 14: Performance and fairness of non-linear models

Columns left to right: Test performance, train performance and difference between fairness on training data and fairness on test data The thresholded neural network again dominates on training data but is surpassed by the adversarial model on test data for higher values of t. The suprising exception is the downsampled adult data set. Results for the credit data sets were omitted as adversarially constrained and variational fair autoencoder models did almost always collapse to degenerate classifiers.

G.2 Mutual information

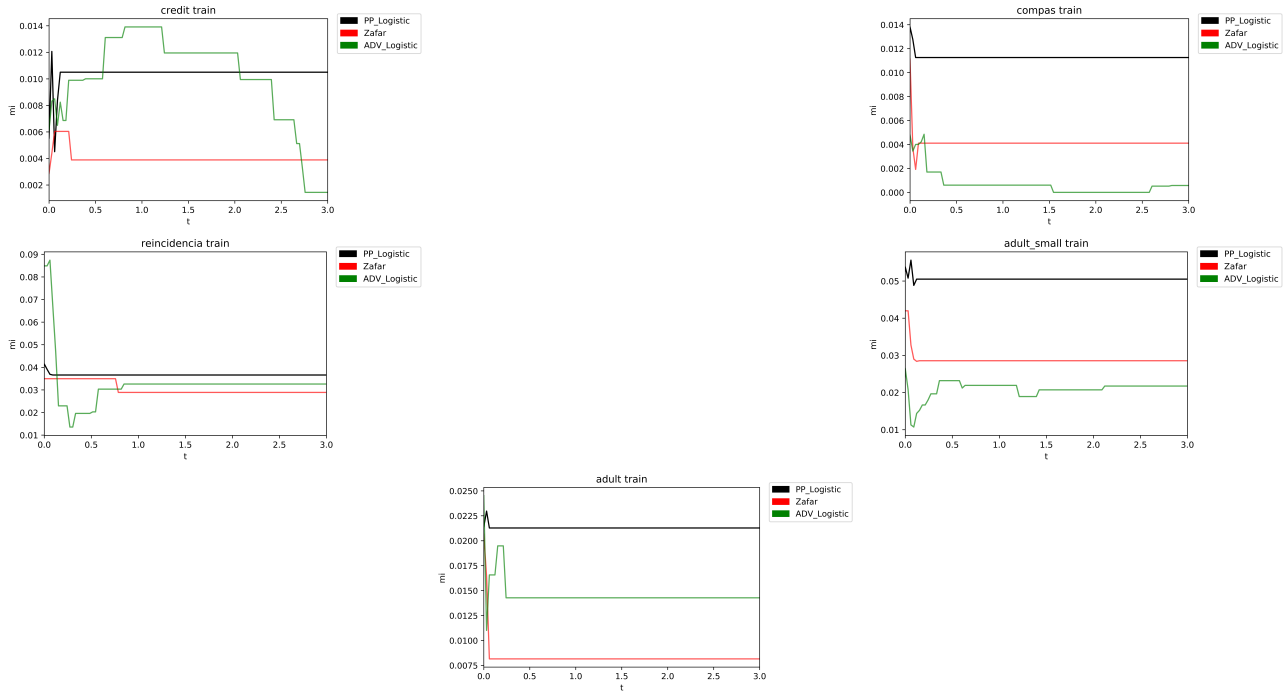

Figure 15: MI Margin distribution and s of Logistic Regression models

Plots show an estimate of mutual information between the margins and s.

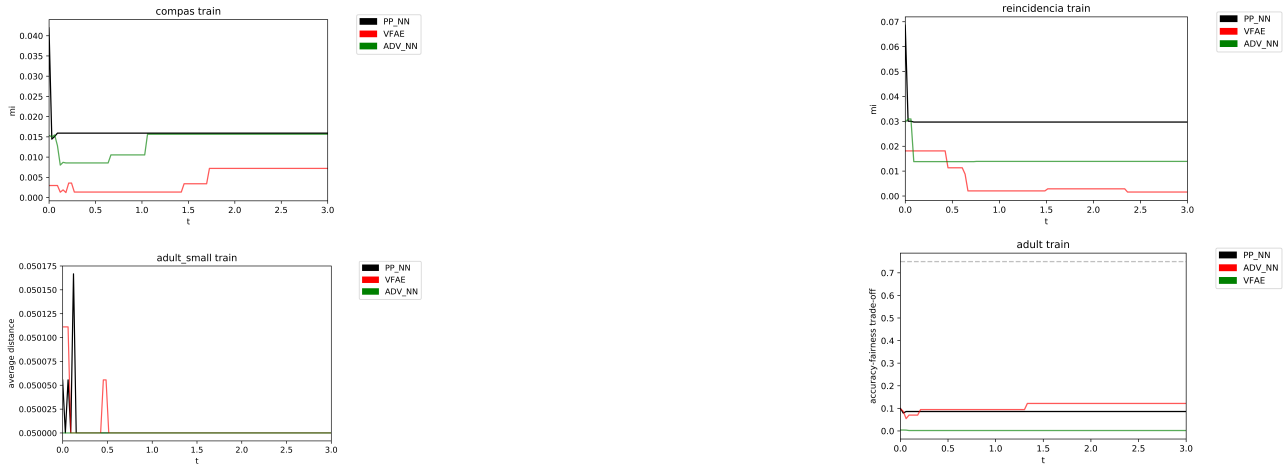

Figure 16: MI Margin distribution and s of neural network models

Plots show an estimate of mutual information between the margins and s.

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Humboldt-Universität zu Berlin
Sprach- und literaturwissenschaftliche Fakultät

Name: Willeke Vorname: Janek

Matrikelnummer: 556286

Eidesstattliche Erklärung zur

☐	**Hausarbeit**	☐	**Bachelorarbeit**
☐	**Take Home-Klausur**	☒	**Masterarbeit**
☐	**Portfolio**	☐	**Sonstiges**

Ich erkläre ausdrücklich, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel

Fairness by Thresholding?

um eine von mir selbstständig und ohne fremde Hilfe verfasste Arbeit handelt. Sie wurde bisher nicht für andere Prüfungen eingereicht.

Ich erkläre ausdrücklich, dass ich *sämtliche* in der oben genannten Arbeit verwendeten fremden Quellen, auch aus dem Internet (einschließlich Tabellen, Grafiken u. Ä.) als solche kenntlich gemacht habe. Insbesondere bestätige ich, dass ich ausnahmslos sowohl bei wörtlich übernommenen Aussagen bzw. unverändert übernommenen Tabellen, Grafiken u. Ä. (Zitaten) als auch bei in eigenen Worten wiedergegebenen Aussagen bzw. von mir abgewandelten Tabellen, Grafiken u. Ä. anderer Autorinnen und Autoren (Paraphrasen) die Quelle angegeben habe.

Mir ist bewusst, dass Verstöße gegen die Grundsätze der Selbstständigkeit als Täuschung betrachtet und entsprechend der fachspezifischen Prüfungsordnung und/oder der Fächerübergreifenden Satzung zur Regelung von Zulassung, Studium und Prüfung der Humboldt-Universität (ZSP-HU) geahndet werden.

Datum 05.11.19 Unterschrift