Presentation for *Probably Approximately Metric-Fair Learning*, Extension and Beyond

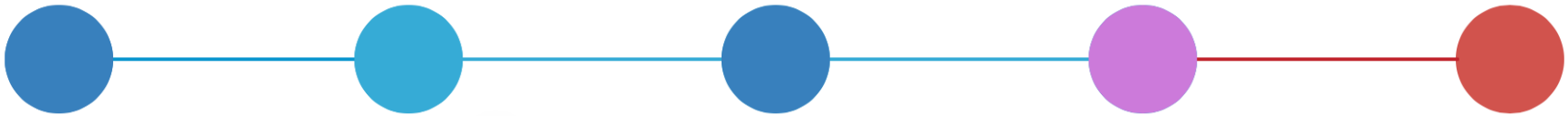
Probably Approximately Metric-Fair Learning is a paper by Rothblum, G. N.,& Yona, G.(2018)

Zhibo Zhang, zhibozhang@cs.toronto.edu
Guided by Prof.Richard Zemel

University of Toronto

December 4, 2019

Overview: How the Story Begins



Problem

Strict Individual Fairness does not Generalize

Solution

Approximate Individual Fairness Using Relaxed Metric

Generalization

- Bound the Loss Using Rademacher Complexity
- Define Relaxed PACF Learning. Apply it to Linear Regression and Logistic Regression.

Thinking

This paper is purely theoretical work.
Are there any practical ways to combine individual fairness and accuracy together in general machine learning?

Extension

- Metric Learning Combined with Revised Siamese Network for Optimizing both Utility and Individual Fairness

1.Individual Fairness

Definition

Suppose we are given:

1. *An input space* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$;
2. *A metric* $\mathcal{D}$ *that defines the similarity between pair of input instances (i.e.* $\mathcal{D}(x_1, x_2)$*);*
3. *A classifier* h *for prediction tasks;*
4. *A distance metric* d *that measures the distance between output pairs (e.g.* $d(h(x_1), h(x_2))$*).*

We say the classifier h ***is individually fair if and only if*** $\forall x_i, x_j \in \mathcal{X}, d(h(x_i), h(x_j)) \leq \mathcal{D}(x_i, x_j)$

Basically, the individual fairness requires the output pairs have distances strictly less than or equal to those of the according inputs.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012, January)

2.Difficulty (Motivation for Approximation): Strict Individual Fairness is Hard to Generalize

- Trivial classifiers, i.e. generates the same outputs for all the input instances, always satisfies individual fairness both on the sample but under the entire distribution, **but no utility in prediction.**
- Suppose the sample is not representative of the underlying data distribution (likely to happen) or some certain representation not included in the sample, the learned classifier won't generalize to the whole distribution in terms of fairness.

3.Approximate Individual Fairness Metric

Definition (loss)

The relaxed fairness loss given a similarity metric Δ and relaxation γ is defined to be:

$$\ell_{\gamma,\Delta}(h,(x_1,x_2)) = \begin{cases} 1, & \text{for } d(h(x_1),h(x_2)) > \Delta(x_1,x_2)+\gamma \\ 0, & \text{for } d(h(x_1),h(x_2)) \leq \Delta(x_1,x_2)+\gamma \end{cases}$$

in which way, the individual fairness loss for the entire distribution is:

$$\mathcal{L}^F_{\mathcal{D},\gamma}(h) = \mathbb{E}_{x_1,x_2\sim\mathcal{D}}[\ell_{\gamma,\Delta}(h,(x_1,x_2))]$$

the empirical loss under sample S is:

$$\hat{\mathcal{L}}^F_{\mathcal{D},\gamma}(h) = \frac{2}{m-1}\sum_{x_1\in S}\sum_{x_2\in S}[\ell_{\gamma,\Delta}(h,(x_1,x_2))]$$

4.PAC Learning

- Generalization error: $\mathcal{L}(h) = \mathbb{E}_{x\sim\mathbb{D}}[\mathbb{1}(h(x) = c(x))]$
  (c is the target concept, h belongs to hypothesis space $\mathcal{H}$)
- Empirical error: $\hat{\mathcal{L}}_S(h) = \frac{1}{m}\sum_{x\in S}\mathbb{1}(h(x) = c(x))$

Definition (PAC Learning)

A concept class $\mathcal{C}$ is said to be PAC-learnable if $\exists$ algorithm $\mathcal{A}$ and a polynomial function poly(,,,) s.t. for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ on input space $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, if $m \geq poly(\frac{1}{\epsilon}, \frac{1}{\delta}, n, size(c))$, there is:

$$\mathbb{P}_{S\sim\mathcal{D}^m}[\hat{\mathcal{L}}_S(h) \leq \epsilon] \geq 1 - \delta$$

$\mathcal{C}$ is said to be efficiently PAC-learnable if $\mathcal{A}$ runs in $poly(\frac{1}{\epsilon}, \frac{1}{\delta}, n, size(c))$

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018)

5.Rademacher Complexity

Definition (Empirical Rademacher Complexity)

For a sample $S = \{z_1, z_2, ..., z_m\}$, the empirical Rademacher Complexity on the hypothesis space $\mathcal{H}$ is defined to be:

$$\hat{\mathcal{R}}_S(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(z_i)\right] = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{\boldsymbol{\sigma}\cdot h(\mathbf{z})}{m}\right]$$

where the general Rademacher Complexity on the distribution $\mathcal{D}$ is the expectation of the above one over the entire distribution: $\mathcal{R}_m(\mathcal{H}) = \mathbb{E}_{S\sim\mathcal{D}^m}\left[\hat{\mathcal{R}}_S(\mathcal{H})\right]$. **The Rademacher complexity measures to what extent the classifiers in the hypothesis space correlate with the random noise σ. Since it takes the supremum over the space, the more complex the space is, the more likely that it contains classifiers that correlate well with the noise.**

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018)

6.Generalization Bounds

In order to show that this approximation generalizes well to the underlying distribution, Rothblum, G. N., & Yona, G. (2018) uses Rademacher complexity to prove a convergence bound for the loss:

Theorem

Let $\mathcal{H}$ be a hypothesis space with Rademacher complexity $R_m(\mathcal{H})$. $\forall \delta, \gamma \in (0,1), \forall G \geq 1, \forall m \geq 0$ (w.o.l.g) assume m is odd, with probability at least $1-\delta$ over an i.i.d sample $S \sim \mathcal{D}^m$, $\forall h \in \mathcal{H}$:

$$\mathcal{M} \leq \hat{\mathcal{L}}^F_\gamma(h) \leq \mathcal{N}$$

where $\mathcal{M} = \mathcal{L}^F_{\gamma+\frac{1}{G}}(h) - 2G(4\hat{R}_{\frac{m-1}{2}}(\mathcal{H}) + \frac{4+17\sqrt{\ln(4/\delta)}}{\sqrt{m-1}})$

and $\mathcal{N} = \mathcal{L}^F_{\gamma+\frac{1}{G}}(h) + 2G(4\hat{R}_{\frac{m-1}{2}}(\mathcal{H}) + \frac{4+17\sqrt{\ln(4/\delta)}}{\sqrt{m-1}})$

7.How to Understand the Generalization Bounds

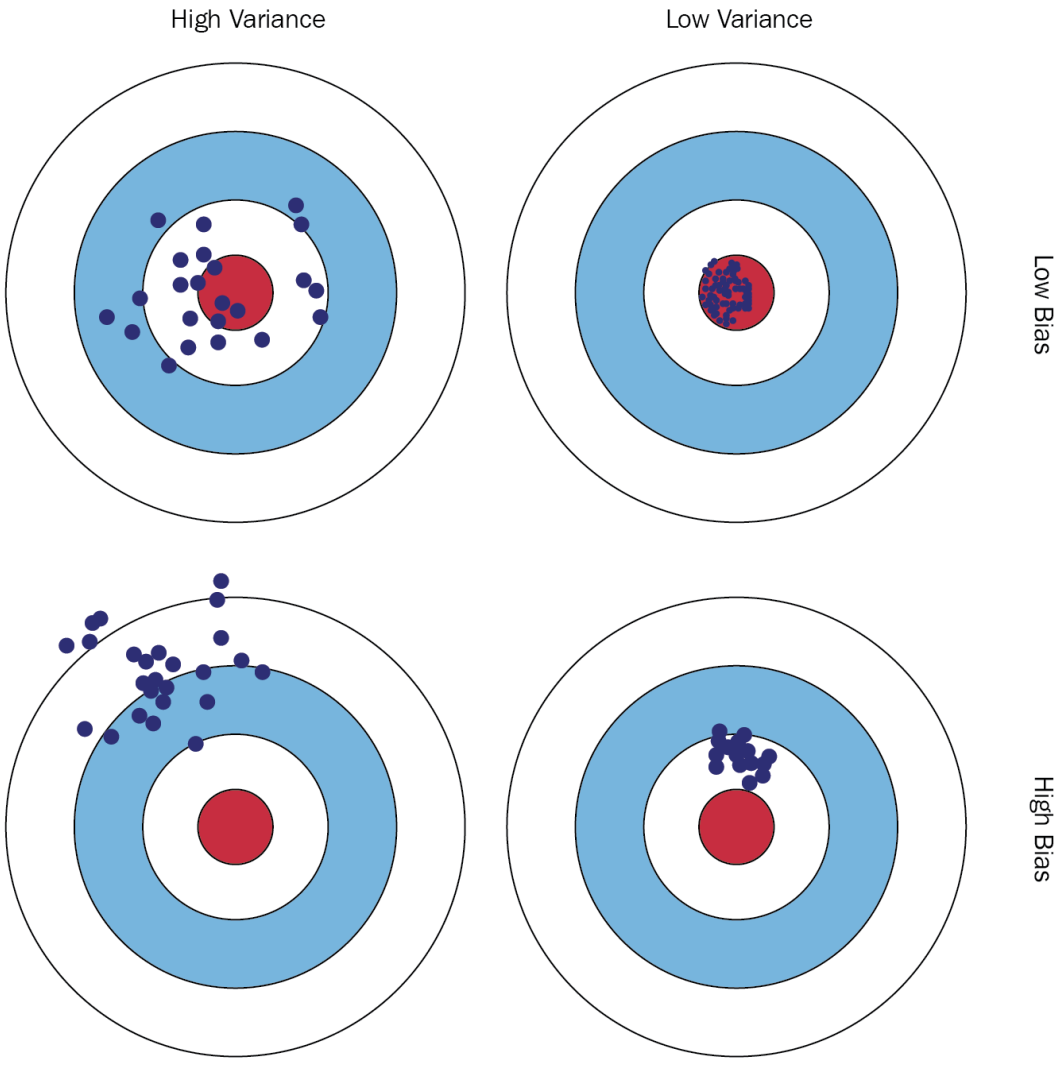
Based on the above bound, there are two essential conditions for a good generalizability:

- The hypothesis space $\mathcal{H}$ has a small Rademacher Complexity on the sample;
- The training sample size m is large enough.

Why should m be large? My understanding:

1. When there is little data in the sample, the learner is likely to choose a classifier that "looks good" based on the performance on little data. But it potentially overfits (low bias and high variance). Intuitively speaking, the learner does not have enough evidence to verify its choice.
2. When there is enough data in the sample (large m), the learner has enough evidence to verify its choice.

8.Visualization



https://www.oreilly.com/library/view/hands-on-transfer-learning/9781788831307/1b3d8196-0617-4e1d-aa79-b2b5ef6f4a8b.xhtml

9.PACF Learnability

Definition

A hypothesis space $\mathcal{H}$ is PACF learnable w.r.t a univariate accuracy loss function $\ell^U : \mathcal{H} \times Z \to \mathbb{R}_+$ and a bivariate fairness loss function $\ell_d^F : \mathcal{H} \times Z \times Z \to \mathbb{R}_+$ if there exists:

- *a function $m_{\mathcal{H}}(\alpha, \gamma, \delta, \epsilon, \epsilon_\alpha, \epsilon_\gamma) : (0,1)^6 \to \mathbb{N}$ (polynomial in $\alpha, \gamma, \log \frac{1}{\delta}, \epsilon, \epsilon_\alpha, \epsilon_\gamma$)*
- *a learning algorithm with the following property: For every required fairness parameters $\alpha, \gamma \in (0,1)$, failure probability $\delta \in (0,1)$ and error parameters $\epsilon, \epsilon_\alpha, \epsilon_\gamma \in (0,1)$ and for every distribution $\mathcal{D}$ over $Z = \mathcal{X} \times \mathcal{Y}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\alpha, \gamma, \delta, \epsilon, \epsilon_\alpha, \epsilon_\gamma)$ i.i.d examples generated by $\mathcal{D}$, the algorithm returns a hypothesis h such that with probability at least $1 - \delta$:*
  - $\mathcal{L}^{Fairness}_{\mathcal{D},\gamma,d}(h) \leq \alpha$, *where* $\mathcal{L}^{Fairness}_{\mathcal{D}}(h) = \mathbb{E}_{z,z' \sim \mathcal{D}}[\ell^F_\gamma(h, z, z')]$
  - $\mathcal{L}^{Accuracy}_{\mathcal{D}}(h) \leq \mathcal{L}^{Accuracy}_{\mathcal{D}}(\mathcal{H}^{\alpha-\epsilon_\alpha,\gamma-\epsilon_\gamma}) + \epsilon$, *where* $\mathcal{L}^{Accuracy}_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell^A_\gamma(h, z)]$

10.Relaxed PACF Learnability

Definition ($g(\cdot)$-relaxed PACF learnability)

A hypothesis space $\mathcal{H}$ is $g(\cdot)$-relaxed PACF learnable w.r.t a function $g : [0,1]^2 \to [0,1]^2$, a univariate accuracy loss function $\ell^U : \mathcal{H} \times Z \to \mathbb{R}_+$ and a bivariate fairness loss function $\ell_d^F : \mathcal{H} \times Z \times Z \to \mathbb{R}_+$ if there exists:

- *a function $m_{\mathcal{H}}(\alpha, \gamma, \delta, \epsilon, \epsilon_\alpha, \epsilon_\gamma) : (0,1)^6 \to \mathbb{N}$ (polynomial in $\alpha, \gamma, \log \frac{1}{\delta}, \epsilon, \epsilon_\alpha, \epsilon_\gamma$)*
- *a learning algorithm with the following property: For every required fairness parameters $\alpha, \gamma \in (0,1)$, failure probability $\delta \in (0,1)$ and error parameters $\epsilon, \epsilon_\alpha, \epsilon_\gamma \in (0,1)$ and for every distribution $\mathcal{D}$ over $Z = \mathcal{X} \times \mathcal{Y}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\alpha, \gamma, \delta, \epsilon, \epsilon_\alpha, \epsilon_\gamma)$ i.i.d examples generated by $\mathcal{D}$, the algorithm returns a hypothesis h such that with probability at least $1 - \delta$:*
  - $\mathcal{L}^{Fairness}_{\mathcal{D},\gamma,d}(h) \leq \alpha$, *where* $\mathcal{L}^{Fairness}_{\mathcal{D}}(h) = \mathbb{E}_{z,z' \sim \mathcal{D}}[\ell^F_\gamma(h, z, z')]$
  - $\mathcal{L}^{Accuracy}_{\mathcal{D}}(h) \leq \mathcal{L}^{Accuracy}_{\mathcal{D}}(\mathcal{H}^{g(\alpha,\gamma)-(\epsilon_\alpha,\epsilon_\gamma)}) + \epsilon$, *where* $\mathcal{L}^{Accuracy}_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell^A_\gamma(h, z)]$

11.PACF Learnability for Linear/Logistic Regression

Now it is time to apply relaxed PACF learnability into linear and logistic classifiers.

1. **In linear settings**, $H_{lin} \stackrel{def}{=} \left\{ \mathbf{x} \to \frac{1+\langle \mathbf{w}, \mathbf{x} \rangle}{2} : ||\mathbf{w}|| \leq 1 \right\}$

Theorem

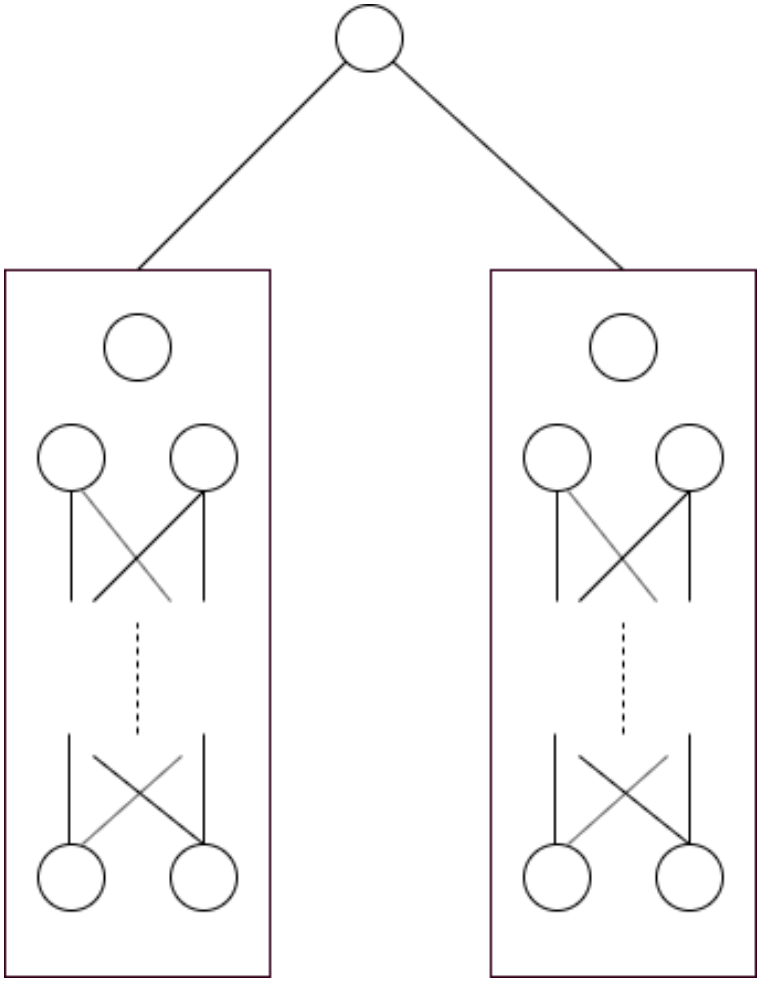
$\forall \gamma^* \in (0,1)$. *$H_{lin}$ is relaxed PACF learnable with* $g(\alpha, \gamma) = (\alpha \cdot \gamma - \gamma^*, \gamma^*)$ *and sample and time complexities of* $poly(\frac{1}{\epsilon_\gamma}, \frac{1}{\epsilon_\alpha}, \frac{1}{\epsilon}, \log \frac{1}{\delta})$

2. **In logistic settings**,
   $H_{\phi,L} \stackrel{def}{=} \{ \mathbf{x} \to \phi_\ell(\langle w, \mathbf{x} \rangle) : \mathbf{w} \in \mathcal{X}, \ell \in [0, L] \}$

Theorem

$\forall L > 0, \forall \gamma^* \in (0,1)$, *$H_{lin}$ is relaxed PACF learnable with* $g(\alpha, \gamma) = (\alpha \cdot \gamma - \gamma^*, \gamma^*)$, *with sample and time complexities that are polynomial in* $poly(\frac{1}{\epsilon_\gamma}, \frac{1}{\epsilon_\alpha}, \frac{1}{\epsilon}, \log \frac{1}{\delta})$

Extension: 1.Embedding Individual Fairness Loss into General ML Training Objective



Formulation:
Given data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$,
$\forall(\mathbf{x}_i, \mathbf{x}_j) \in M(\mathcal{D})$,
$h_1(\mathbf{x}_i) = \hat{y}_i, h_2(\mathbf{x}_j) = \hat{y}_j$

1. Utility Loss:
   $\mathcal{L}_i(h_1(\mathbf{x}_i), y_i), \mathcal{L}_j(h_2(\mathbf{x}_j), y_j)$

2. Individual Fairness Loss:
   $[d(h_1(\mathbf{x}_i), h_2(\mathbf{x}_j)) - \Delta(\mathbf{x}_i, \mathbf{x}_j)]$
   $\mathbb{1}(d > \Delta)$

For every possible pair, we back-propagate for both accuracy and individual fairness loss.

2.How to Get the Metric?

Right now, we are facing two problems:

1. How do we acquire the individual fairness metric Δ for measuring the input distances?
2. If we have some subjective individual fairness notions available, can we elicit it into this model?

- Yes, both can be solved using metric learning.

Suppose we are given some subjective notions of individual fairness notions, i.e. $\forall \mathbf{x}_i \in \mathcal{D}$, there is an according label that is marked manually by experts (different from y_i), then we can feed the instances and their labels into the general metric learning like large margin nearest neighbors[1].

Question:

How is this architecture different from the Siamese Network?

[1] Weinberger, K. Q., & Saul, L. K. (2009).

Algorithm 1: Training process for the individually fair and accurate learning

Feed the training instances $\mathcal{X}$ and the according subjective labels for individual fairness into the LMNN learning, get the distance metric Δ;

for $1 \leq i \leq |\mathcal{X}|$ **do**

$\mathcal{X}' = \mathcal{X}[i:] + \mathcal{X}[:i]$

1. **for** $1 \leq epoch \leq 10$ **do**

   Feed $\mathcal{X}, \mathcal{X}'$ into the architecture proposed. Calculate the prediction losses for sub-networks $\mathcal{A}$ and $\mathcal{B}$ separately and perform back-propagation.

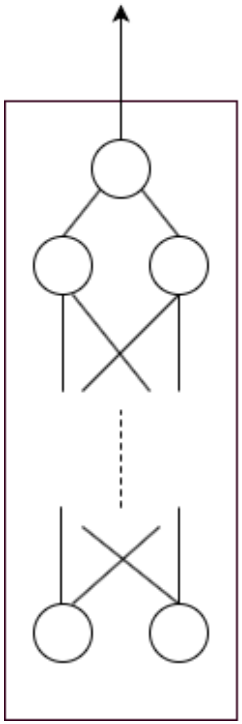
   end

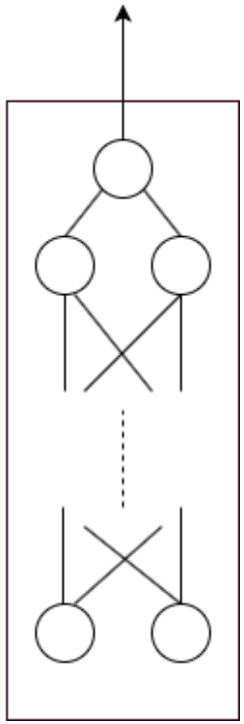
2. Feed $\mathcal{X}, \mathcal{X}'$ into the architecture proposed again.

3. Run through $\mathcal{A}$ and $\mathcal{B}$, calculate the similarity loss and perform back-propagation.

end

3.Alternative Structure





In the case where the fairness notions are hard binary labels: Just use two separate models. When optimizing the weights relative to loss, swap the outputs of the two individuals that are supposed to be treated similarly and then do back-propagation. This is suitable for the case that the subjective notions are binary.

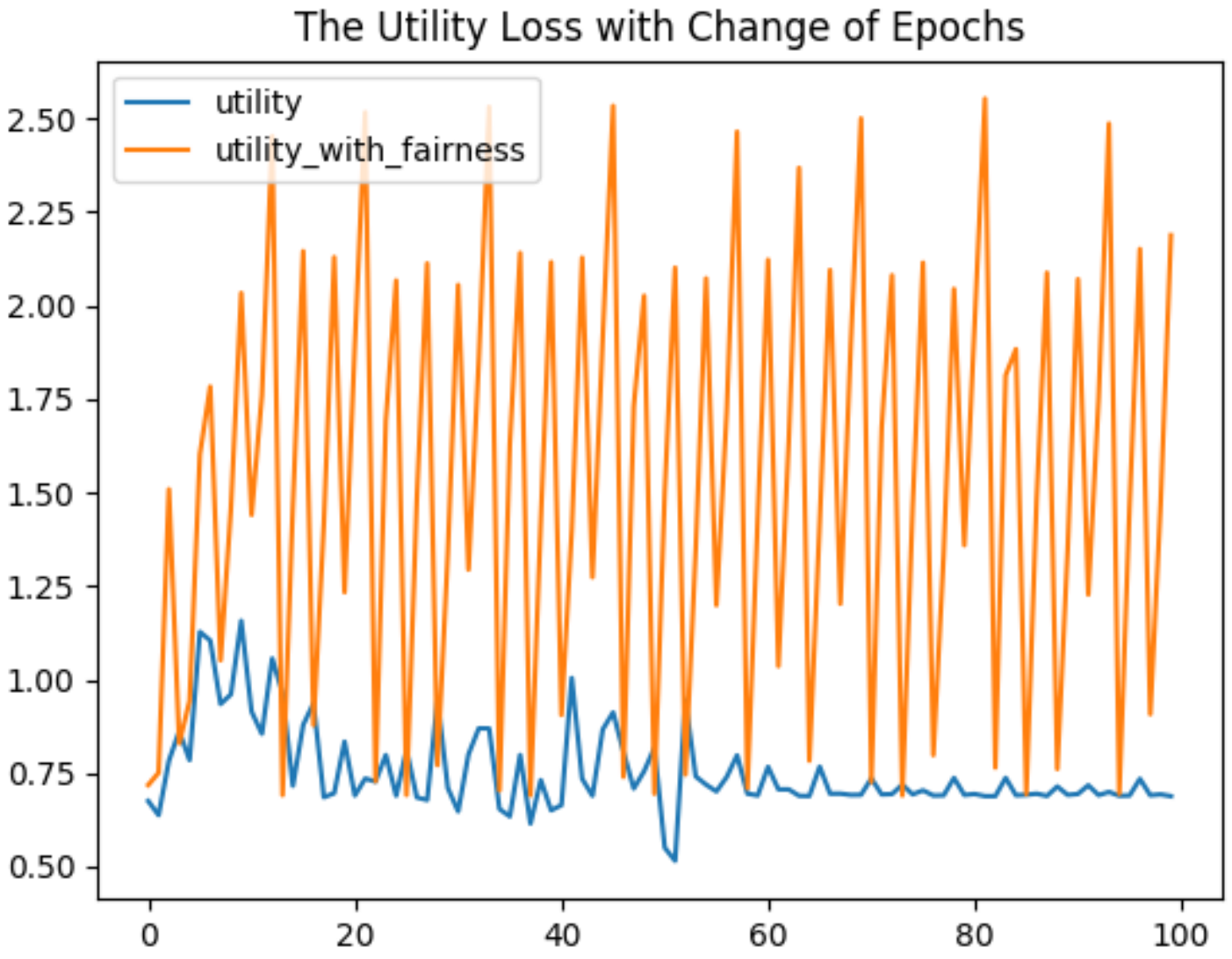
4.Experiment

- **Data Set**: Compas Recidivism Data [2]- compas-scores-two-years. I use the binary values on whether the individuals commit a crime or not in the next few years as the labels. The rest of the features are inputs.
- **Simulation of the Notions of the Subjective Individual Fairness:** I generate a series of 0s and 1s for the individuals. If a pair of individual have the same notion, they are supposed to be treated similarly in terms of the machine prediction task.
- I define the individual fairness loss in the following way:

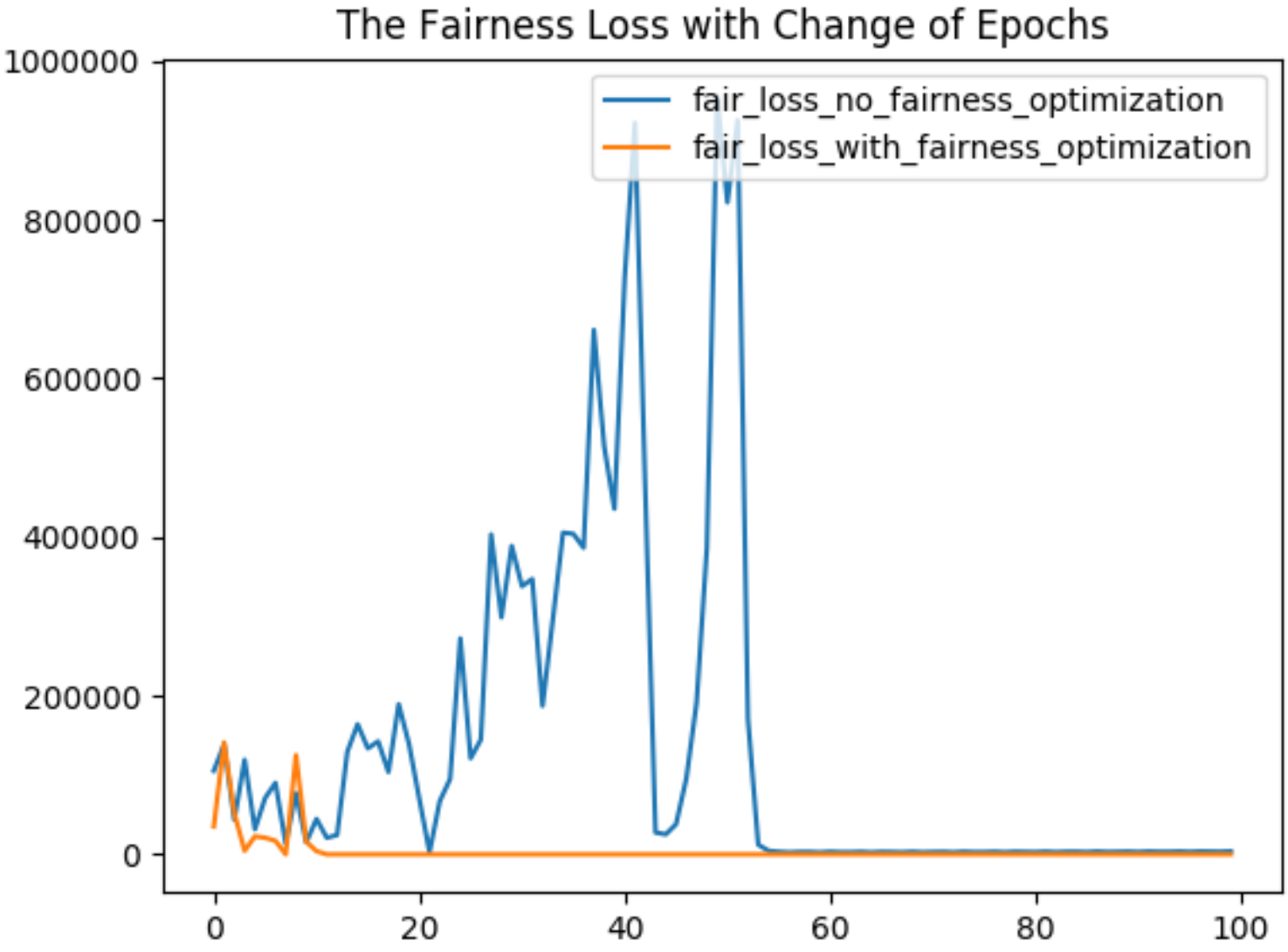
$$\mathcal{L}^F = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} [(\hat{y}_i - \hat{y}_j)^2] \mathbb{1}(i \text{ and } j \text{ should be treated similarly})$$

[2]https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis

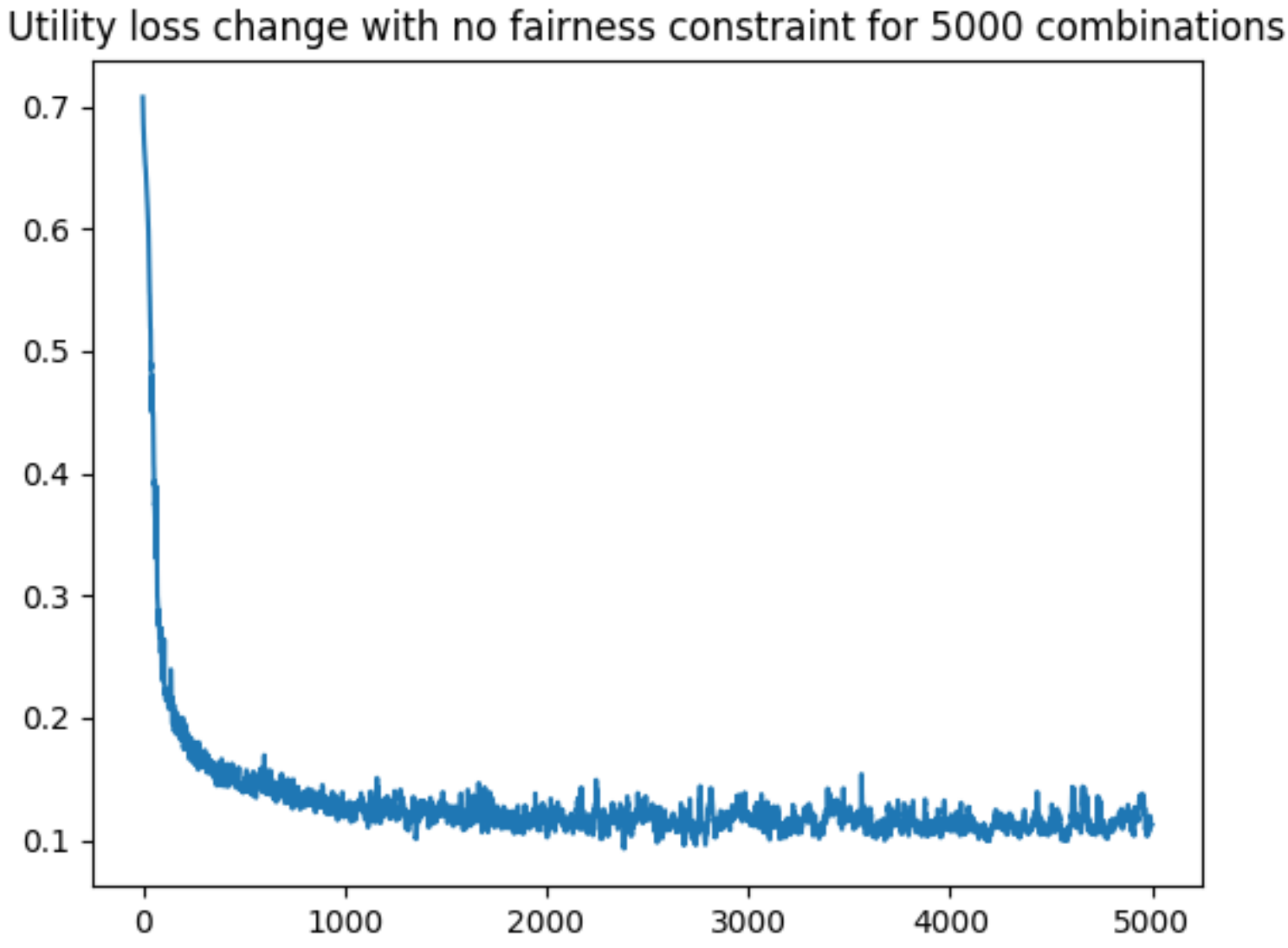
5.Experimental Results



6.Experimental Results



7.Experimental Results



References

- Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012, January). Fairness through awareness. In Proceedings of the 3rd innovations in theoretical computer science conference (pp. 214-226). ACM.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. MIT press.
- Rothblum, G. N., & Yona, G. (2018). Probably approximately metric-fair learning. arXiv preprint arXiv:1803.03242.
- Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 10(Feb), 207-244.
- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In Advances in neural information processing systems (pp. 737-744).