Fast & Fair: Efficient Second-Order Robust Optimization for Fairness in ML

Allen Minch[1], Hung Anh Vu[2], Annie Warren[3], advised by Dr. Elizabeth Newman[4]

[1] Brandeis University, [2] University of Maryland, [3] University of Minnesota, [4] Emory University

Abstract.

This project explores adversarial training techniques to develop fairer Deep Neural Networks (DNNs) to mitigate the inherent bias they are known to exhibit. DNNs are susceptible to inheriting bias with respect to sensitive attributes such as race and gender, which can lead to life-altering outcomes (e.g., demographic bias in facial recognition software used to arrest a suspect). We propose a robust optimization problem to improve fairness in DNNs, and leveraging second-order information, we are able to efficiently find a solution.

Adversarial training

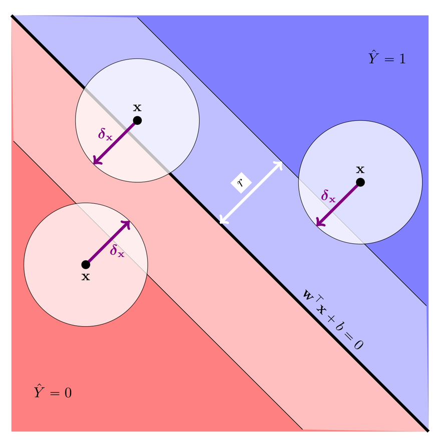


Figure 1: Robust optimization

Our research investigates whether implementing robust optimization - so that two nearby points are more likely to be classified similarly - would improve fairness. Robust optimization takes into consideration at a radius r around a data point (see fig. 1).

Optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \left[\max_{\|\boldsymbol{\delta_x}\|\le r} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta_x}),\boldsymbol{y})\right] + R(\boldsymbol{\theta})$$

Solving the inner optimization problem

Projected Gradient Descent: We use a first-order method with iterates

$$\boldsymbol{\delta}_{\boldsymbol{x}}^{(k+1)} = P\left[\boldsymbol{\delta}_{\boldsymbol{x}}^{(k)} + \alpha^{(k)} \cdot \nabla_{\boldsymbol{x}} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta_x}),\boldsymbol{y})\right]$$

where P is a projection operator such that the constraint, $\|\boldsymbol{\delta_x}\| \le r$, is satisfied.

Trust Region Subproblem: We use a second-order Taylor approximation of our loss function and solve

$$\boldsymbol{\delta_x}(\lambda) = -(\nabla_{\boldsymbol{x}}^2 L(f_{\boldsymbol{\theta}}(\boldsymbol{x}),\boldsymbol{y}) + \lambda \boldsymbol{I})^{-1} \nabla_{\boldsymbol{x}} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}),\boldsymbol{y})$$

where $\lambda \ge 0$ is chosen such that the constraint is satisfied.

Error in trust region subproblem solution

For binary classification, we use a logistic regression loss function with a sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$, so we have the following inner optimization problem:

$$\max_{\|\boldsymbol{\delta_x}\|\le r} \left[-y \ln\big(\sigma(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\delta_x})+b)\big) - (1-y)\ln\big(1-\sigma(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\delta_x})+b)\big)\right]$$

When we use a second-order approximation (RHS) instead of the true loss function (LHS), we end up solving two slightly different problems with differing terms

$$\sigma(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\delta_x})+b) \ne \sigma(\boldsymbol{w}^\top\boldsymbol{x}+b) + \sigma'(\boldsymbol{w}^\top\boldsymbol{x}+b)\boldsymbol{w}^\top\boldsymbol{\delta_x}$$

Taylor expanding the LHS, we can obtain the following:

$$\sigma(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\delta_x})+b) \approx \sigma(\boldsymbol{w}^\top\boldsymbol{x}+b) + \sigma'(\boldsymbol{w}^\top\boldsymbol{x}+b)\boldsymbol{w}^\top\boldsymbol{\delta_x} + \frac{1}{2}\boldsymbol{\delta_x}^\top \boldsymbol{w}\sigma''(\boldsymbol{w}^\top\boldsymbol{x}+b)\boldsymbol{w}^\top\boldsymbol{\delta_x} + \dots$$

The error of solving the trust region subproblem comes from truncating the higher-order terms in the Taylor expansion. The error from truncating the quadratic term depends on the magnitude $|\sigma''(z)| \le 0.1$ and the $\boldsymbol{\delta_x}$ for which we solved.

Fairness metrics

Let $\hat{Y}$ be the classifier's prediction, Y be the true class, and s be a binary sensitive attribute. How do we measure the **fairness** of a binary classifier with respect to s?

Independence:	$\mathbb{P}(\hat{Y}=1 \mid s=0) = \mathbb{P}(\hat{Y}=1 \mid s=1)$
Separation:	$\mathbb{P}(\hat{Y}=1 \mid Y=1, s=0) = \mathbb{P}(\hat{Y}=1 \mid Y=1, s=1)$
Sufficiency:	$\mathbb{P}(Y=1 \mid \hat{Y}=1, s=0) = \mathbb{P}(Y=1 \mid \hat{Y}=1, s=1)$

Hiring (synthetic data)

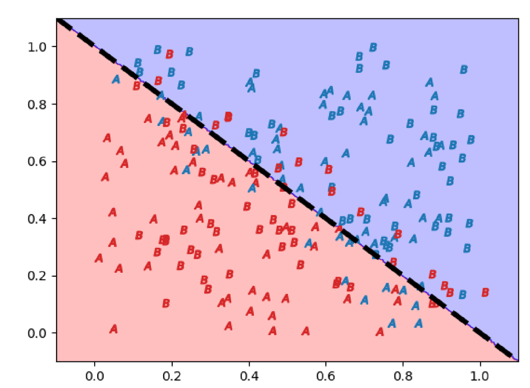


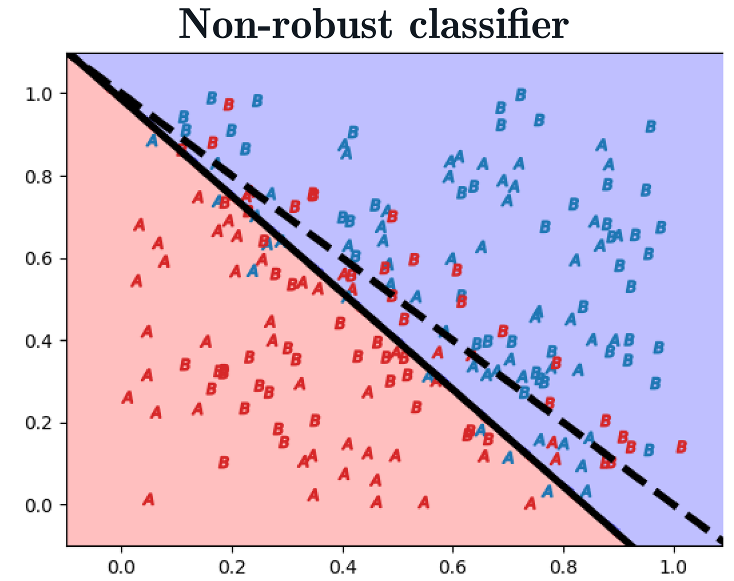
Figure 2: Unfair setup of data

- Y: should be hired or not hired
- $\hat{Y}$: predicted to be hired or not hired
- s: sensitive attribute A or B
- **unfairness:** Bs shifted up and right ↗
  As shifted down and left ↙

All red individuals in the blue region who would be hired in error are Bs, while all blue individuals in the red region who would be incorrectly not hired are As.

Hiring results

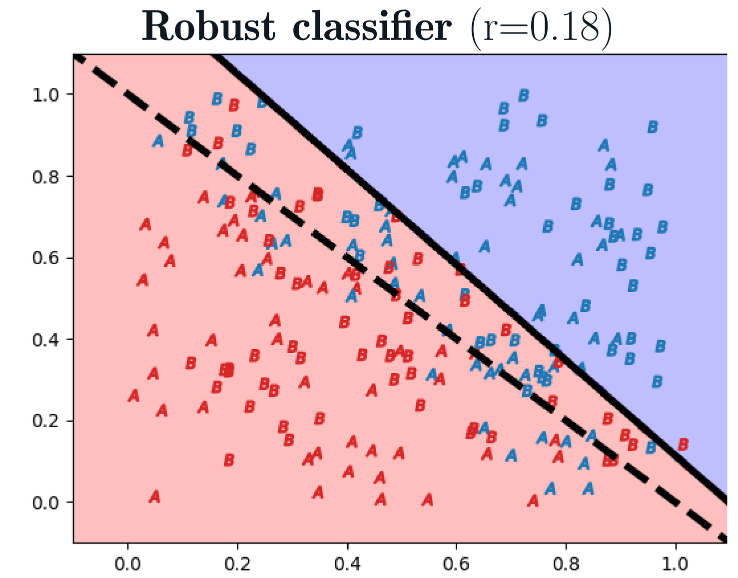
Non-robust classifier



Diff:	Y=0 |S1-S0|	Y=1 |S1-S0|
Ind.	0.152	0.152
Sep.	0.248	0.179
Suff.	0.213	0.190

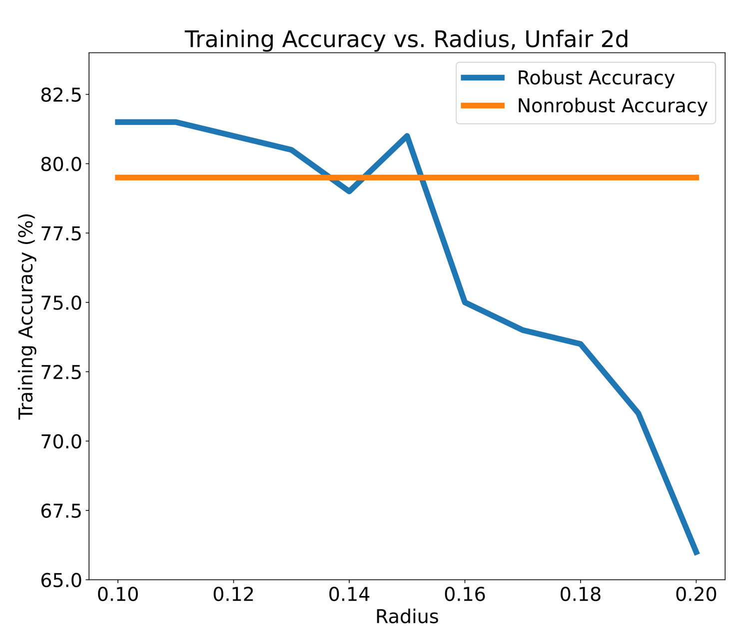
Training Accuracy: 79.5%
Test Accuracy: 78.0%

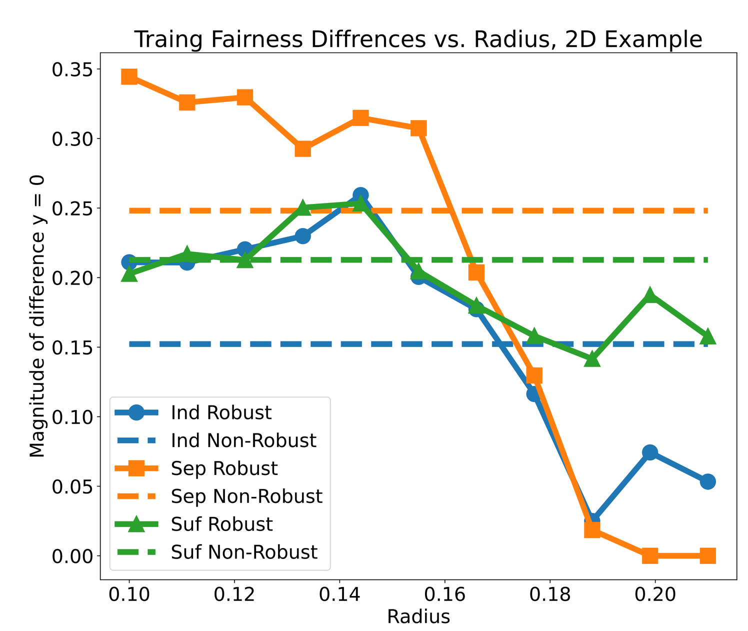
Robust classifier (r=0.18)



Diff:	Y=0 |S1-S0|	Y=1 |S1-S0|
Ind.	0.025	0.025
Sep.	0.019	0.127
Suff.	0.142	0.038

Training Accuracy: 73.5%
Test Accuracy: 73.0%



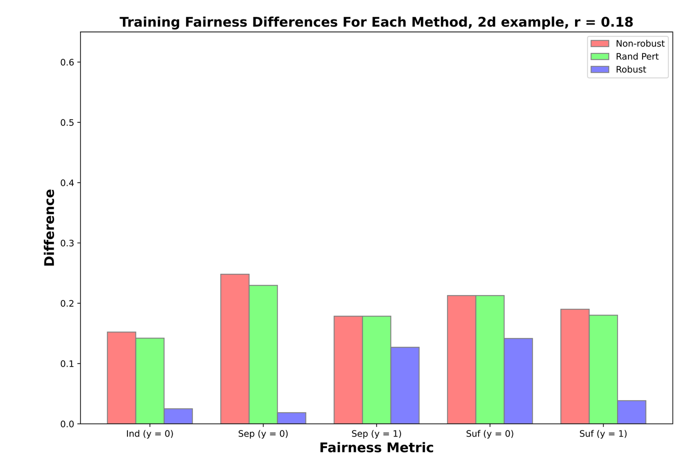


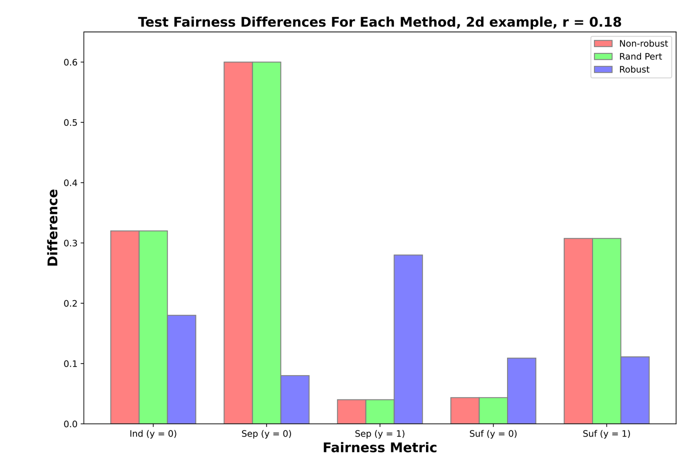
Which method is faster?

Avg. Epoch Time $\frac{PGD}{Trust}$		
Dataset:	**Min**	**Max**
Synthetic	1.377	3.130
LSAT	2.852	9.639
Adult	8.497	31.407

Unlike in PGD, where we may have to do many gradient computations, with the second-order method, we at worst solve a system of linear equations and use a bisection method. The trust region method has proven to be the more efficient method.

Which method is fairer?





Conclusions

- Utilizing the trust region subproblem method **significantly improves efficiency**: computing second-order information using `hessQuik` outperforms first-order PGD across all perturbation radii on three different data sets.
- **Robustness can improve fairness, but potentially at the cost of accuracy.** Fairness improves as the perturbation radius increases but accuracy decreases in both training and testing data as the radius increases.
- If using robust training with a certain radius improves fairness, it appears to improve fairness by larger margins compared to random perturbation; **solving the optimization problem well is worthwhile.**

Acknowledgements

- Thank you to Dr. Elizabeth Newman, our mentor, for her guidance and support.
- This work is supported in part by the US NSF award DMS-2051019.

References

[1] Martin Arjovsky et al. *Invariant Risk Minimization.* 2020.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities.* http://www.fairmlbook.org. fairmlbook.org, 2019.

[3] Elizabeth Newman and Lars Ruthotto. "'hessQuik': Fast Hessian computation of composite functions". In: *Journal of Open Source Software* 7.72 (2022), p. 4171.

[4] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* 2e. New York, NY, USA: Springer, 2006.

[5] Tai Le Quy et al. "A survey on datasets for fairness-aware machine learning". In: *WIREs Data Mining and Knowledge Discovery* 12.3 (2022).