Improving Fairness Generalization Through a Sample-Robust Optimization Method

Julien Ferry[1], Ulrich Aïvodji[2], Sébastien Gambs[3], Marie-José Huguet[1] and Mohamed Siala[1]

[1]LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
[2]École de Technologie Supérieure, Montréal, Canada
[3]Université du Québec à Montréal, Montréal, Canada

January 3, 2023

Table of Contents

Context

- The paper is published in the **Machine Learning** journal, in July 2022
- Link to the paper `link.springer.com/article/10.1007/s10994-022-06191-y`
- Preprint: `https://hal.archives-ouvertes.fr/hal-03709547`
- Open source code `https://github.com/ferryjul/FairnessSampleRobustness`

- Fairness in machine learning
- Statistical measures
- Generalisation of fairness on unseen data is one of the open challenges for trustworthy machine learning

The COMPAS Example [Angwin et al., 2016]

- Binary classification task: Recidivism within two years
- Sensitive attribute: Ethnicity (African-American/Caucasian)
- Protected Groups:
  - $\mathcal{A}$: African-American individuals;
  - $\mathcal{B}$: Caucasian individuals;

Statistical Fairness

- Principle: ensure that some measure $\mathcal{M}$ *differs by no more than* ϵ between several *protected* groups
- In the particular case of two protected groups ($\mathcal{A}$) and ($\mathcal{B}$), one need to ensure that $|\mathcal{M}(\mathcal{A}) - \mathcal{M}(\mathcal{B})| < \epsilon$

Supervised Fair Learning: A Bi-Objective Optimization Problem

- Notation: $\mathcal{D}$ initial dataset, h prediction model, ϵ unfairness tolerance
- Let unf$(\cdot)$ be an unfairness oracle. A common formulation of the Fair Learning problem is:

$$\begin{aligned} \underset{h \in \mathcal{H}}{\arg\min} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \text{unf}(h, \mathcal{D}) \leq \epsilon \end{aligned} \tag{1}$$

  where one wants to build model h minimizing objective function f_{obj} and exhibiting unfairness withing an ϵ threshold (on training dataset $\mathcal{D}$)
- The fairness constraint does not generalize well in practice [Cotter et al., 2018, 2019]
- Existing approaches are essentially adhoc without a theoretical framework [Cotter et al., 2018, 2019; Chuang and Mroueh, 2021; Huang and Vishnoi, 2019; Mandal et al., 2020; Sagawa et al., 2019; Taskesen et al., 2020; Wang et al., 2021]

Distributionally Robust Optimization (DRO)

- Instead of minimizing objective function f_{obj} for a given distribution $\mathcal{P}$, DRO aims at minimizing f_{obj} for a worst-case distribution among a *perturbation set* of $\mathcal{P}$ [Sagawa et al., 2019] $\mathcal{B}(\mathcal{P})$

Perturbation Set based on the Jaccard Distance

- Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two sample sets. The Jaccard distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as follows: $J_\delta(\mathcal{D}_1, \mathcal{D}_2) = 1 - \frac{|\mathcal{D}_1 \cap \mathcal{D}_2|}{|\mathcal{D}_1 \cup \mathcal{D}_2|}$
- Let $a \in [0, 1]$, we define a perturbation set $\mathcal{B}(\mathcal{D}, a)$ as the set of subsets of $\mathcal{D}$ whose Jaccard distance from the $\mathcal{D}$ is less than or equal to a.

Example Perturbation Sets Structure

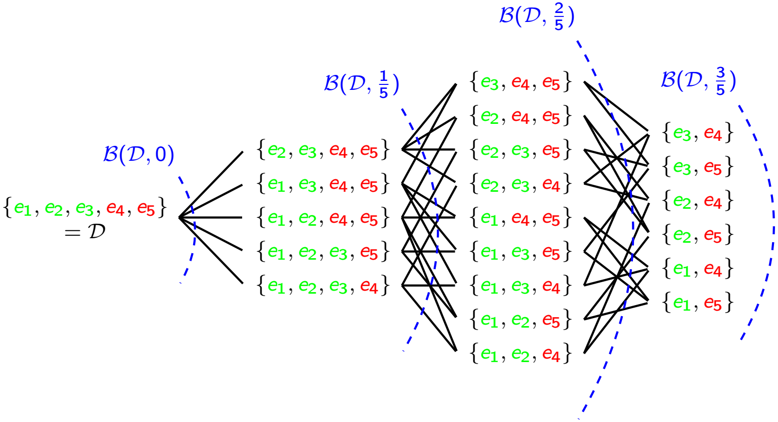
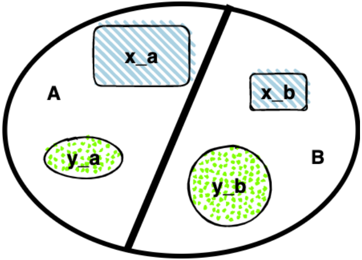


Figure: Example of perturbation sets for a dataset $\mathcal{D}$ with 5 examples and two protected groups a ($\{e_1, e_2, e_3\}$) and b ($\{e_4, e_5\}$). Subsets that can not be used to audit a model's fairness with respect to protected groups a and b are not represented.

- In order to evaluate the robustness of a given model, we need an efficient way to find the largest perturbation set $\mathcal{P}$ such that h is fair on each element in $\mathcal{P}$
- The value corresponding to this set is denoted by $\mathcal{SR}(h, \mathcal{D}, \epsilon)$
- The sample-robust fair learning problem is such that

$$\begin{aligned} \arg\min_{h \in \mathcal{H}} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \mathcal{SR}(h, \mathcal{D}, \epsilon) > d \end{aligned} \tag{2}$$

- To build a Pareto frontier, start with $d = 0$, then at each iteration increase the value of d with respect to the last solution
- Instead of using brute force, we propose an integer programming approach to find the robustness value $\mathcal{SR}(h, \mathcal{D}, \epsilon)$
- We show that our approach is flexible and efficient in practice
- For the sake of scalability, we also propose a linear time greedy approach



- Let $\mathcal{D}'$ be the subset of $\mathcal{D}$ where the colored sets are removed
- If h is not fair on $\mathcal{D}'$, then h is not robust on the perturbation set defined with the distance between $\mathcal{D}$ and $\mathcal{D}'$

The Integer Program

$$\min \quad n \tag{3}$$

$$\text{s.t.} \quad n = x_a + x_b + y_a + y_b \tag{4}$$

$$\left|\frac{M_a - x_a}{N_a - x_a - y_a} - \frac{M_b - x_b}{N_b - x_b - y_b}\right| > \epsilon \tag{5}$$

$$0 \leq x_a \leq M_a \qquad 0 \leq x_b \leq M_b$$

$$0 \leq y_a \leq M_a - M_a \qquad 0 \leq y_b \leq N_b - M_b$$

$$x_a + y_a < N_a \qquad x_b + y_b < N_b$$

- The optimal value can be used to identified the largest perturbation set where the robustness constraint is satisfied

Table of Contents

- Instead of ensuring fairness on each subset of $\mathcal{D}$ up to Jaccard distance d, we enforce the fairness constraint on a number of randomly generated subsets $\mathcal{B}_\omega(\mathcal{D})$ including $\mathcal{D}$
- Our formulation of the Heuristic Sample-Robust Fair Learning problem is:

$$\begin{aligned} \underset{h \in \mathcal{H}}{\arg\min} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \forall \mathcal{D}' \in \mathcal{B}_\omega(\mathcal{D}),\ \text{unf}(h, \mathcal{D}') \leq \epsilon \end{aligned} \tag{6}$$

Table of Contents

Setup description

- We compare:
  - The original `FairCORELS` [Aïvodji et al., 2019]
  - The exact approach with and without validation set
  - The heuristic approach with 10 and 30 subsets
- Four fairness metrics:
  - Statistical Parity [Dwork et al., 2012]
  - Predictive Equality [Chouldechova, 2017]
  - Equal Opportunity [Hardt et al., 2016]
  - Equalized Odds [Hardt et al., 2016]
- Four biased datasets:
  - Adult Income dataset [Frank and Asuncion, 2010]
  - COMPAS dataset [Angwin et al., 2016]
  - Default Credit dataset [Yeh and Lien, 2009]
  - Bank Marketing dataset [Moro et al., 2014]
- A wide range of unfairness tolerances
- The Integer Program is solved using the constraint programming solver OrTools

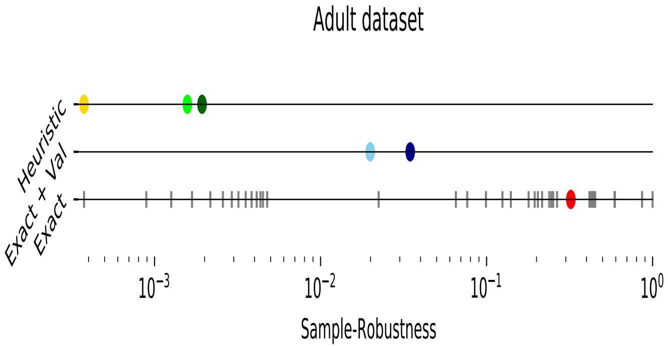


Figure: Fairness sample-robustness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Statistical Parity metric, $\epsilon = 0.01$) for the adult dataset

Exact vs. Heuristic Methods II

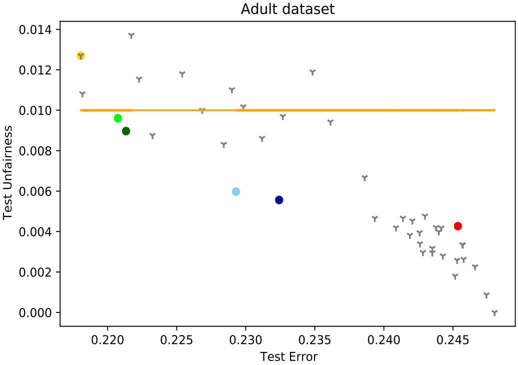


Figure: Test error and unfairness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Statistical Parity metric, $\epsilon = 0.01$)

TensorFlow Constrained Optimization

- TensorFlow Constrained Optimization[a] (TFCO) is a Python library for optimizing inequity-constrained problems in TensorFlow to produce machine learning models (not restricted to the fair learning problem)
- Implementing our heuristic sample-robust fair method into TFCO simply requires declaring additional constraints (one per protected group per subset)

[a] https://github.com/google-research/tensorflow_constrained_optimization

Compared Methods

We build on the setup of Cotter et al. [2019] and compare the following approaches:

- `unconstrained` trains is the default model without enforcing fairness constraints
- `baseline` is the standard fair learning approach
- `validation` is a state-of-the-art approach proposed in Cotter et al. [2018, 2019] to improve fairness generalization.
- `dromasks-n` is the integration of our method into `baseline`, using *n* subset (in practice, we use $\mathtt{n} \in \{10, 30, 50\}$).

Setup

- We run four distinct experiments using different fairness metrics, datasets (including numerical features) and (non-binary) sensitive attributes

	Proxy Lagrangian				Lagrangian			
	Train		Test		Train		Test	
Model	Error	Viol.	Error	Viol.	Error	Viol.	Error	Viol.
Adult Income Dataset								
`unconstrained`	.122	.072	**.144**	.071	.122	.072	**.144**	.071
`baseline`	.141	0	*.154*	.009	.141	0	*.155*	.006
`validation`	.132	-.002	.158	.004	.134	0	.157	*.004*
`dromasks-10`	.14	-.003	.156	*.003*	.143	-.001	*.155*	**-.003**
`dromasks-30`	.14	-.004	.157	**-.001**	.148	-.002	.156	**-.003**
`dromasks-50`	.14	-.003	.157	**-.001**	.151	-.002	.157	**-.003**

Table: Results of the experimental study of the heuristic approach using `TFCO` (error rates and maximum fairness violations)

Table of Contents

Summary

- We address the problem of fairness generalisation via an approach based on Distributionally Robust Optimization
- Our propositions are flexible enough to be used with different models (even black box models)
- We propose exact and heuristic methodologies
- We empirically show that our approach is competitive to the state-of-the-art with two learning models on many datasets in the literature using different fairness measures

Future Work

- Can we find an efficient way to approximate the best parameters ?
- How to extend the work with other distance functions ?

Thank you!

References I

Aïvodji, U., Ferry, J., Gambs, S., Huguet, M.-J., and Siala, M. (2019). Learning fair rule lists. *arXiv preprint arXiv:1909.03977*.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. (2017a). Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 35–44. ACM.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. (2017b). Learning certifiably optimal rule lists for categorical data. *The Journal of Machine Learning Research*, 18(1):8753–8830.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias. *ProPublica, May*, 23.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163.

Chuang, C.-Y. and Mroueh, Y. (2021). Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*.

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

Cotter, A., Gupta, M., Jiang, H., Srebro, N., Sridharan, K., Wang, S., Woodworth, B., and You, S. (2019). Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *International Conference on Machine Learning*, pages 1397–1405. PMLR.

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.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California. *School of information and computer science*, 213:2–2.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*.

Huang, L. and Vishnoi, N. (2019). Stable and fair classification. In *International Conference on Machine Learning*, pages 2879–2890. PMLR.

Mandal, D., Deng, S., Jana, S., Wing, J., and Hsu, D. J. (2020). Ensuring fairness beyond the training data. *Advances in Neural Information Processing Systems*, 33.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

Rivest, R. L. (1987). Learning decision lists. *Machine learning*, 2(3):229–246.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Taskesen, B., Nguyen, V. A., Kuhn, D., and Blanchet, J. (2020). A distributionally robust approach to fair classification. *arXiv preprint arXiv:2007.09530*.

Wang, Y., Nguyen, V. A., and Hanasusanto, G. A. (2021). Wasserstein robust support vector machines with fairness constraints. *arXiv preprint arXiv:2103.06828*.

Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

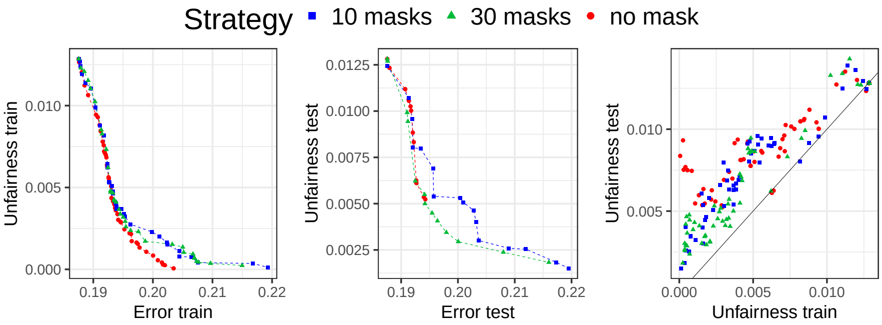


Figure: Results obtained on the Default Credit dataset, for the Predictive Equality metric

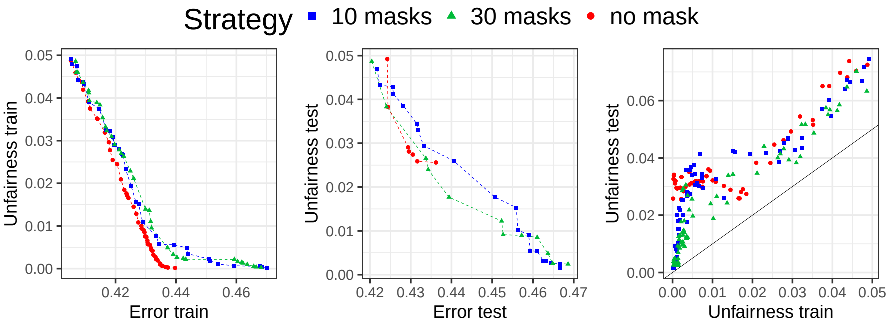


Figure: Results obtained on the COMPAS dataset, for the Statistical Parity metric

Rule Lists: Definition

Rule lists [Rivest, 1987] are classifiers formed by an ordered list of *if-then* rules with antecedents in the *if* clauses and predictions in the *then* clauses. More precisely, a rule list $r = (\{p_{k,k\in\{1..K\}}\}, \{q_{k,k\in\{1..K\}}\}, q_0)$ consists of K distinct association rules $p_k \rightarrow q_k$, in which p_k is the antecedent of the association rule and q_k its associated consequent, followed by a default prediction q_0.

A possible rule list for the example dataset of slide ?? (with 100% accuracy)

```
if [Education:Dropout] then [low]
else if [Gender:Male AND Age>45] then [high]
else [low]
```

FairCORELS Problem Formulation

- Based on the CORELS algorithm [Angelino et al., 2017a,b]
- FairCORELS [Aïvodji et al., 2019] returns rule list r^* that is a solution to the following problem:

$$\begin{aligned} \arg\min_{r \in \mathcal{R}} \quad & \text{misc}(h, \mathcal{D}) + \lambda . K_r \\ \text{s.t.} \quad & \text{unf}(h, \mathcal{D}) \leq \epsilon \end{aligned}$$

where:

- $\mathcal{R}$ is the space of rule lists
- $\mathcal{D}$ denotes the training dataset
- K_r is the length of rule list r
- λ is a regularization parameter balancing sparsity and accuracy
- misc$(\cdot)$ is the misclassification error and unf$(\cdot)$ measures unfairness

FairCORELS search space

- FairCORELS represents the search space of rule lists as a prefix tree (trie)
- FairCORELS leverages several bounds to efficiently explore this search space (including CORELS' original bounds)

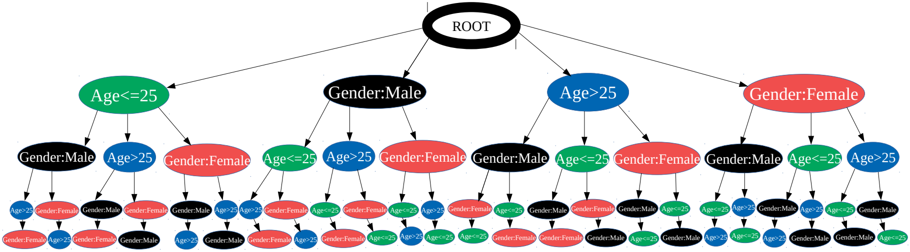


Figure: Example prefix tree with 4 attributes