FairCORELS, an Open-Source Library for Learning Fair Rule Lists

Ulrich Aïvodji[1], Julien Ferry[2], Sébastien Gambs[1], Marie-José Huguet[2] and Mohamed Siala[2]
[1]UQAM, Montréal, Canada [2]LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France

Fairness in Machine Learning

- FairCORELS uses **statistical fairness** notions [5]
- Objective: ensure that some statistical measure m of a classifier's h outputs has equal (or close) value between several *protected* subgroups
  eg. with two protected groups A and B, ensure that:
  $$|m(h, A) - m(h, B)| \leq \epsilon$$
- Different metrics proposed, differing on the statistical measure to be equalized accross protected groups

Rule Lists

- **Rule lists** [4] are classifiers formed by an ordered list of *if-then* rules with antecedents in the *if* clauses and predictions in the *then* clauses, followed by a default prediction

Example rule list

```
if [gender:Female] then [high]
else if [Age<=25] then [low]
else [high]
```

- Rule lists are **inherently interpretable models** that exhibit interesting properties for interpretability [3]

FairCORELS: Learning Certifiably Optimal Fair RulE Lists

- CORELS [2] is a **branch-and-bound algorithm** to build certifiably optimal (in terms of accuracy/sparsity) rule lists r^*. It explores the search space of rule lists $\mathcal{R}$ and solves the following minimization problem:
  $$r^* = \arg\min_{r \in \mathcal{R}} \ \texttt{misc}(r, X, Y) + \lambda . K_r$$
  where X and Y are the training instances unprotected features and labels (respectively), K_r denotes the length of rule list r, and $\texttt{misc}(r, X, Y)$ is the training classification error. The λ hyperparameter controls the accuracy/sparsity tradeoff
- FairCORELS [1] is a **multi-objective variant of CORELS**, designed to learn fair rule lists. It returns rule list r^* minimizing CORELS's objective function, and exhibiting *fairness* at least ϵ (*unfairness* at most $1 - \epsilon$). Formally, FairCORELS solves the following problem:
  $$r^* = \arg\min_{r \in \mathcal{R}} \quad \texttt{misc}(r, X, Y) + \lambda . K_r$$
  $$\text{s.t.} \quad \texttt{unf}(r, X, Y, A) \leq 1 - \epsilon,$$
  where A defines the protected subgroups, $(1 - \epsilon)$ is the fairness tolerence, and $\texttt{unf}(r, X, Y, A)$ is the training fairness violation
- Similar to CORELS, FairCORELS represents the search space of rule lists $\mathcal{R}$ using a **prefix tree**. It leverages several bounds and proposes a collection of exploration strategies (BFS, DFS, Best-First searches...) to efficiently explore this search space

Example Use of FairCORELS

- Example Python code to train a rule list with statistical parity fairness constraint

```
# Create the classifier object
clf = FairCorelsClassifier(
        n_iter=2.5*1e6, # Max. size of the prefix tree
        c=10e-3,  # Regularization parameter Lambda
        policy="bfs", # Exploration heuristic
        fairness=1, # 1 for Statistical Parity
        epsilon=epsilon, # The fairness constraint
        maj_vect=A_train_u, # Binary vector (protected group 1)
        min_vect=A_train_p) # Binary vector (protected group 2)
# Train it
clf.fit(
    X_train,
    y_train,
    features=featuresNames,
    prediction_name="high_income")
```

Learning Sets of Accuracy/Fairness Tradeoffs

- FairCORELS can easily be used to produce different tradeoffs between accuracy and fairness, by varying the ϵ parameter

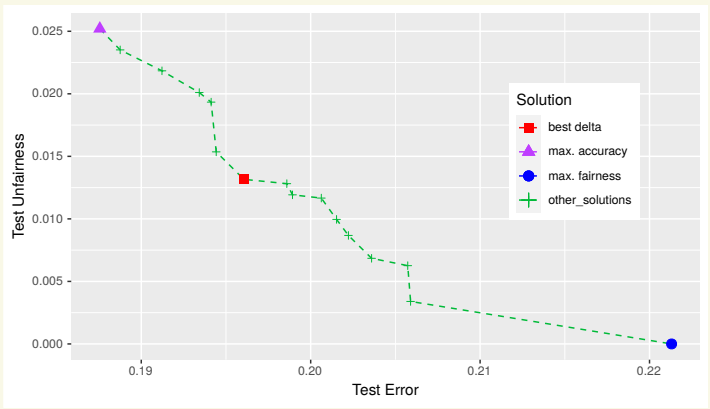

Figure: Example tradeoffs produced by FairCORELS, for the Statistical Parity fairness metric, on the Default of Credit Card dataset

Getting Started with the FairCORELS Open-Source Python Library

- Our source code is available on GitHub, along with example scripts and notebooks: https://github.com/ferryjul/fairCORELS
- It is based on the CORELS algorithm [2] and its original[1]and Python implementation[2]
- FairCORELS is also available on PyPI[3], which allows for an easy install with `pip install faircorels`

[1]https://github.com/corels/corels, [2]https://github.com/corels/pycorels, [3]https://pypi.org/project/faircorels

References

[1] Ulrich Aïvodji et al. "Learning fair rule lists". In: *arXiv preprint arXiv:1909.03977* (2019).
[2] Elaine Angelino et al. "Learning certifiably optimal rule lists for categorical data". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 8753–8830.
[3] Alex A Freitas. "Comprehensible classification models: a position paper". In: *ACM SIGKDD explorations newsletter* 15.1 (2014), pp. 1–10.
[4] Ronald L Rivest. "Learning decision lists". In: *Machine learning* 2.3 (1987), pp. 229–246.
[5] Sahil Verma and Julia Rubin. "Fairness Definitions Explained". In: *IEEE/ACM International Workshop on Software Fairness* 18 (2018).

Julien Ferry (PhD candidate)
jferry@laas.fr - https://homepages.laas.fr/jferry/
LAAS-CNRS