Automatic Protected Attributes Partial Correlation Removal for Fairness Tabular Data Science

Shaked Cohen and Ori Cohen*

[1] Bar Ilan University, Ramat Gan, Israel
shkcohen@gmail.com
[2] Bar Ilan University, Ramat Gan, Israel
oricohen70@gmail.com

Abstract. Fairness in machine learning has become a critical issue in recent years, as models can unintentionally discriminate against certain groups of people[6]. To address this issue, various fairness definitions have been proposed[2], and numerous methods have been developed to achieve fairer models[1]. In this paper, we propose a novel solution to improve fairness through correlation removal using partial correlation. Our proposed method involves using partial correlation to calculate the correlation between each feature and the sensitive feature while controlling for other features. We evaluate our approach on four real-world datasets, demonstrating that our method can balance the performance-fairness trade-off and add another preprocessing tool to the data science toolbox for building fairer models. Our results suggest that our method can be effective in mitigating discriminatory effects while retaining high performance, and thus, it can help to promote fairness in machine learning applications.

* Supported by Cohen Industries

Problem Description

Bias and fairness are important topics in data science, particularly in the context of machine learning. Bias can be defined as a systematic error in the modeling process that leads to incorrect predictions for a particular group of individuals. This bias can be introduced in many different ways, such as the collection of biased data or the selection of biased features. One of the most concerning issues in machine learning is unfairness, which can result from biased data or models that produce discriminatory predictions based on sensitive attributes such as race, gender, or age. Unfairness in machine learning can have real-world consequences, such as discrimination in hiring practices or loan approvals. The causes of bias and unfairness in machine learning are complex and can vary depending on the context. For example, biased data can be a result of historical or societal factors, such as under-representation of certain groups in a particular dataset. Biased features can also be introduced inadvertently, such as using a variable that is correlated with the outcome variable but also correlates with a sensitive attribute, leading to unfair predictions. Research shows that even if protected attributes (such as race a sex) are dropped entirely from training data, unfairness and bias towards these groups can still occur. Often the reason is "proxy features" - seemingly neutral variables that contain "clues" that lead back to protected attributes. It is important to address bias and fairness in machine learning to ensure that models are both accurate and equitable. A variety of methods have been developed to address these issues, including preprocessing, post-processing, and in-processing techniques. In our work, we aim to mitigate unfairness using a preprocessing technique - correlation removal. Therefore, the DS element we wish to improve is preprocessing. Although, the real "target" we wish to improve is essentially the result of the trained model on evaluation metrics - both fairness and general performance. Specifically, our goal is to mitigate unfairness in terms of:

- Demographic parity difference
- Equalized odds difference

While minimizing the degradation in performance, in terms of:

- Accuracy
- Roc Auc score

These metrics were chosen arbitrarily and for simplicity, however further research should address additional metrics as well. It is worth mentioning that fairness has many definitions, often contradicting each other, and in some cases in order to answer a certain fairness criteria the model can't have optimal performance (on an unfair dataset) by definition.

Solution Overview

Before introducing our approach, let us introduce the two baselines we compare our solution to, and then explain the novelty in our solution.

Baseline 1. Original data, vanilla model (Logistic Regression) This method represents the vanilla fairness-unaware model. Although other methods don't introduce any fairness awareness or constraint, they do apply preprocessing that indirectly influences model fairness. Therefore, this is the most performance oriented model.

Baseline 2. Correlation removal through linear transformations, vanilla model (Logistic Regression) In this method data is preprocessed using fairlearn's CorrelationRemover transformer - applies a linear transformation to the non-sensitive feature columns in order to remove their correlation with the sensitive feature columns while retaining as much information as possible (as measured by the least-squares error). This method will change the original dataset by removing all correlation with sensitive values. Correlation threshold α can be tweaked for softer affect. Mathematically: let's assume in the original dataset $\mathbf{X}$ we've got a set of sensitive attributes $\mathbf{S}$ and a set of non-sensitive attributes $\mathbf{Z}$. This method will be solving the following problem:

$$\min_{\mathbf{z}_1,\ldots,\mathbf{z}_n} \sum_{i=1}^{n} \|\mathbf{z}_i - \mathbf{x}_i\|^2 \, subjectto \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \left(\mathbf{s}_i - \overline{\mathbf{s}}\right)^T = \mathbf{0}$$

The solution to this problem is found by centering sensitive features, fitting a linear regression model to the non-sensitive features and reporting the residual. This baseline represents a preprocessing correlation removal approach previously introduced in the literature, **having fairness in mind**.

Our Proposed Solution - Partial Correlation Removal

Instead of removing all correlations between sensitive attributes and features, partial correlation removes the effect of all other features on the correlation between the sensitive attribute and the target variable. The idea behind this approach is that some features may be directly related to the sensitive attribute and removing their correlation may lead to information loss. By using partial correlation, we can preserve the information related to these features while still removing the indirect effects of other features on the correlation. In other words, we can **preserve the information related to sensitive features that may be directly related to the target variable, while still removing the indirect effects of other features on the correlation.** This approach can potentially improve fairness metrics while minimizing information loss. The partial correlation between X and Y controlling for Z is denoted as $X \perp\!\!\!\perp Y \mid Z$. The correlation remover in Fairlearn[3] works by setting a threshold value for the absolute correlation between each feature and the sensitive attribute, and removing any features with correlations higher than the threshold. This approach can remove all correlations between the sensitive attribute and the remaining features, but it can also remove information that is directly related to the target variable. On the other hand, using partial correlation removes only the indirect effects of other features on the correlation between the sensitive attribute and the target variable, while preserving the direct effects of sensitive features on the target variable. This approach can potentially improve fairness while minimizing information loss, and it can be a useful alternative to the Fairlearn correlation remover in cases where we want to preserve some information related to the sensitive attribute.

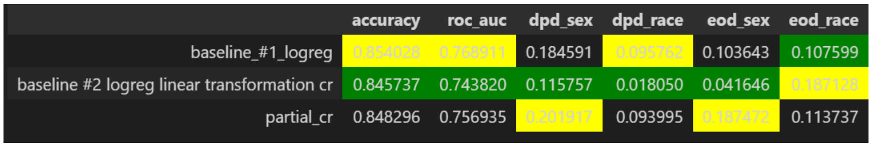
Regarding the sensitive features themselves, I see two options: Either they stay in the training data, and thus affect the model directly, or they are dropped from the training data entirely. We chose to drop them in our experiments, however further work or in practice it may be worth trying both methods and comparing results.

Solution Overview

We evaluate our solution on datasets that are known to have fairness issues and sensitive attributes: Income Dataset UCI, . . .
In the experiments,

1. A dataset with known fairness issue is fetched
2. Mandatory preprocessing in order to be able to fit a model
3. A feature correlation matrix with heatmap style is plotted
4. A dendrogram is also plotted in cases where there are many binary features
5. Sensitive features are dropped from data (but saved for later analysis)
6. Logistic Regression model is fitted on data* and evaluated
7. *Data - 1. original data, 2. data after correlation removal through linear transformations, 3. data after partial correlation removal
8. Fairness metrics are evaluated on the three trained models using the previously saved sensitive features
9. All evaluation metrics are saved and displayed in a table, with yellow and green indicating max and min value of this metric

Results. For the Adult Income Dataset UCI,



	accuracy	roc_auc	dpd_sex	dpd_race	eod_sex	eod_race
baseline_#1_logreg	0.854028	0.768911	0.184591	0.095762	0.103643	0.107599
baseline #2 logreg linear transformation cr	0.845737	0.743820	0.115757	0.018050	0.041646	0.187128
partial_cr	0.848296	0.756935	0.201917	0.093995	0.187472	0.113737

Fig. 1. Adult Income UCI Evaluation. Having eod for equalized odds difference, dpd for demographic parity difference, cr for correlation remover

Results Interpretation. It is interesting to see that our approach with partial correlation remover gives 2nd performance result compare to our baselines. However, it gives 2nd results for fairness metrics as well. Hence, our approach can be used as a balancer for the performance-fairness tradeoff. Of course, in other datasets results may vary, and we believe our method can outperform other methods in some cases. Overall, we can add this approach to the data science toolbox.

Another lead on improving performance can be tweaking the threshold of the partail correlation in order to decide which features to drop. Partial correlation removal has a threshold correlation for feature removal. This threshold should be treated as a hyperparameter, and can be tuned accordingly.
Note: We built our PartialCorrelationRemover class as a sklearn-compatible transformer, and therefore it can be placed in a pipeline, its hyperparameter can be searched for using hyperparameters search such as random search etc.

Code. Code can be found at:

Related Work

In recent years, there has been growing concern about unfairness[9] in machine learning models. Various methods have been proposed to mitigate this issue, including algorithmic fairness[7], pre-processing[4], post-processing[8], and adversarial techniques[5].

One popular approach is to use pre-processing techniques to remove the correlation between the sensitive attribute and the other features in the dataset. One such method is the Fairlearn correlation remover, which sets a threshold value for the absolute correlation between each feature and the sensitive attribute and removes any features with correlations higher than the threshold. While this approach can remove all correlations between the sensitive attribute and the remaining features, it can also remove information that is directly related to the target variable.

Our solution differs in that it uses partial correlation removal, which removes only the indirect effects of other features on the correlation between the sensitive attribute and the target variable, while preserving the direct effects of sensitive features on the target variable. This approach can potentially improve fairness while minimizing information loss.

In addition to these techniques, there has been growing interest in evaluating the fairness of machine learning models. Several metrics have been proposed, including the demographic parity, equal opportunity, and equalized odds. These metrics can be used to measure how much different groups are affected by a model's predictions and to evaluate the effectiveness of fairness mitigation techniques.

Conclusion

An extremely important note regarding the proposed solution and baseline 2 correlation removal, is that **these preprocessing steps best fit a generalized linear model**, where the model learns only linear relationships. Since correlation is a linear statistic in its essence, correlation removal is prone to assist only for these models.
Our findings indicate that our proposed solution shows potential as an additional tool in the DS toolbox for navigating the performance-fairness tradeoff.

Although there is no one-size-fits-all solution to fairness mitigation in machine learning models, each approach has its own strengths and weaknesses. It is recommended to experiment with multiple methods to find the one that works best on your specific dataset and evaluation metrics, as well as hyperparameters tuning.

In this project, we experimented with both research and practice, having the chance to explore various datasets, perform preprocessing pipeline, deal with severe class imbalance (and walk out of it alive), performing automatic feature engineering, diving in statistics - correlation, partial correlation, reading literature on a hot new topic - fairness, getting to know additional non trivial libraries - pingouin statistical package, fairlearn for fairness evaluation and mitigation, writing sklearn-compatible transformer, how to plot correlation when there are many binary variables (dendrogram) and even how to edit the style of a dataframe (coloring the background of chosen cells).

References

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

[2] Alex Beutel et al. "Putting Fairness Principles into Practice: Challenges, Metrics, and Improvements". In: 2019. URL: `https://arxiv.org/pdf/1901.04562.pdf`.

[3] Sarah Bird et al. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Tech. rep. MSR-TR-2020-32. Microsoft, May 2020. URL: `https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/`.

[4] Michael Feldman et al. *Certifying and removing disparate impact*. 2014. DOI: `10.48550/ARXIV.1412.3756`. URL: `https://arxiv.org/abs/1412.3756`.

[5] Xiaoxiao Li et al. *Estimating and Improving Fairness with Adversarial Learning*. 2021. DOI: `10.48550/ARXIV.2103.04243`. URL: `https://arxiv.org/abs/2103.04243`.

[6] Ninareh Mehrabi et al. *A Survey on Bias and Fairness in Machine Learning*. 2019. DOI: `10.48550/ARXIV.1908.09635`. URL: `https://arxiv.org/abs/1908.09635`.

[7] Dana Pessach and Erez Shmueli. *Algorithmic Fairness*. 2020. DOI: `10.48550/ARXIV.2001.09784`. URL: `https://arxiv.org/abs/2001.09784`.

[8] Felix Petersen et al. *Post-processing for Individual Fairness*. 2021. DOI: `10.48550/ARXIV.2110.13796`. URL: `https://arxiv.org/abs/2110.13796`.

[9] Robert C. Williamson and Aditya Krishna Menon. "Fairness risk measures". In: *International Conference on Machine Learning*. Submission to ICML 2019. 2019.