FairTrader: A Method to Smoothly Control the Performance-Fairness Trade-Off

Kiara Stempel^{1,*}, Stefan Kramer¹

¹Johannes Gutenberg University Mainz, Staudingerweg 9, 55128 Mainz, Germany

Abstract

Fair algorithms are a prerequisite if we want to increase user trust, but are meanwhile also required for legal reasons. Increasing the fairness of a model can lead to a decrease in prediction accuracy. While recent work on fairness is often based on developing models that optimize fairness metrics while keeping the reduction in accuracy to a minimum, few focus on how to make this fairness-performance trade-off as controllable as possible. Several methods include a parameter γ for specifying the level of fairness penalty, leading not necessarily to a smooth trade-off curve. By this we mean a curve that deviates only slightly from being monotonically non-increasing. In this paper, we show that by only giving the sample weights generated by an AdaBoost-based variant that is predicting the sensitive attribute to another classifier, discriminative predictions can be reduced and the trade-off can even be made more controllable. Thus, our method allows for the possibility of creating a more strategic balance between fairness and performance of the model through user control.

Keywords

Discrimination, performance-fairness trade-off, classification, boosting

Methods for fair machine learning often struggle to control the trade-off between prediction performance and fairness. Several methods include a parameter γ for specifying the level of fairness penalty [1, 2, 3]. However, varying γ does not necessarily lead to a smooth trade-off curve, i.e., a curve that deviates only slightly from being monotonically non-increasing.

In this paper, we propose FairTrader, a method that makes the progression of the trade-off humanpredictable and thereby more controllable. Given n instances, our method calculates instance weights w_j , j = 1, ..., n using an AdaBoost-based procedure. AdaBoost is a classical boosting algorithm that combines multiple weak learners c_t , t = 1, ..., k to create a strong classifier by focusing on misclassified instances, i.e. it results in an ensemble h [4]. This ensemble is trained on predicting the sensitive attribute s in such a way that instances for which the sensitive attribute s can actually be predicted well from the other features are given a low weight, while instances where s cannot be predicted well are given a high weight. Other methods adapting AdaBoost focus directly on adjusting the weights such that the prediction is directly improved for fairness while training on the target variable y [5, 6, 7]. Subsequently, the returned weights are given to another ensemble model f predicting the actual target. Thus, f can be trained based on these weights in order to learn preferentially from the instances where there is only little correlation between s and other features. For calculating the weights, we adopt the algorithm structure from AdaBoost, but customize the update function of the instance weights as

$$w_j = \left(1 - \frac{c_t(x_j) \cdot s_j}{2}\right)^t \cdot w_j,\tag{1}$$

where t corresponds to the current iteration of AdaBoost. We hand over the sample weights w of all iterations of h as well as the model weights α of all weak learners to the subsequent ensemble f. Following this, the training process of f consists of fitting model d_t , t = 1, ..., k on w_t and assigning weights $1 - \alpha$ to its ensemble members.

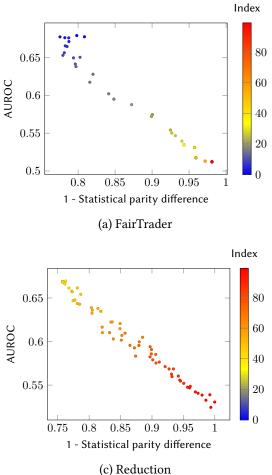
For evaluation, we perform a 15-times hold-out in the outer loop, splitting the data randomly into $\frac{2}{3}$ of training data and $\frac{1}{3}$ of test data each time. Selected hyperparameters are then optimized on the training data using three-fold cross validation in an inner loop. We optimize all models on the area

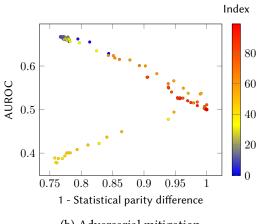
Discovery Science - Late Breaking Contributions 2024

^{*}Corresponding author.

Stempel@uni-mainz.de (K. Stempel); kramer@informatik.uni-mainz.de (S. Kramer)

^{😨 🛈 © 2024} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).





(b) Adversarial mitigation

Table 1: Values of AUTOC and the range that is explored by the corresponding trade-off curve for our method and comparative methods.

Data	Method	AUTOC	$x_{max} - x_{min}$	x_{min}	x_{max}	y_{min}	y_{max}
Compas	FairTrader	0.10	0.21	0.77	0.98	0.51	0.65
Compas	Adversarial	0.08	0.24	0.76	1.0	0.38	0.67
Compas	Reduction	0.09	0.23	0.76	0.99	0.52	0.67
German	FairTrader	0.05	0.09	0.90	0.99	0.50	0.59
German	Adversarial	0.03	0.04	0.94	0.98	0.63	0.71
German	Reduction	0.02	0.05	0.95	1.0	0.66	0.69
Adult	FairTrader	0.10	0.33	0.64	0.97	0.54	0.77
Adult	Adversarial	0.01	0.02	0.95	0.97	0.50	0.54
Adult	Reduction	0.04	0.10	0.82	0.92	0.67	0.70

Figure 1: We compare the trade-off curves of the methods in terms of how much the trade-off points follow each other in a monotonically decreasing sequence.

under the trade-off curve (AUTOC), as points should ideally be close to the top-right corner in order to increase performance as well as fairness measures. For this purpose, the curves are truncated at their minimum and maximum x-values, which also determines the extent to which a curve explores the trade-off space. We compare to the adversarial mitigation method of Zhang et al. [8] and to the reduction approach based on Agarwal et al. [1], which are implemented in *fairlearn* [9]. As fairness metric, we consider the statistical parity difference [10], whereas for performance, we measure AUROC.

Taking the best of all models in terms of AUTOC leads to the preliminary results illustrated in Table 1 and Figure 1. We notice that for the best chosen model for the Compas dataset, the points of the adversarial mitigation method lie for the most part slightly above those of our method. However, we do not consider only the points indicating one specific pair of numbers of the trade-off, but take also a look at the progression of the points in the order they were measured when increasing the fairness parameter, here illustrated by colors. We observe that for adversarial mitigation and the reduction approach, it is less clear to estimate where the next calculated pair of fairness and performance measure will actually lie if we run the algorithm with gradually increasing γ , even though γ represents the balance between training towards predicting y and enforcing the fairness constraint. On the other hand, for FairTrader, we can say that the trade-off curve iteratively decreases monotonically in the majority of cases, especially compared to adversarial learning.

References

- A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, H. M. Wallach, A reductions approach to fair classification, in: Proceedings of the 35th International Conference on Machine Learning (ICML), 2018.
- [2] M. Cerrato, M. Köppel, R. Esposito, S. Kramer, Invariant representations with stochastically quantized neural networks, Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
- [3] F. Kamiran, T. Calders, M. Pechenizkiy, Discrimination aware decision tree learning, in: IEEE International Conference on Data Mining, 2010.
- [4] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: A statistical view of boosting, The Annals of Statistics 38 (2000).
- [5] X. Huang, Z. Li, Y. Jin, W. Zhang, Fair-adaboost: Extending adaboost method to achieve fair classification, Expert Systems with Applications 202 (2022).
- [6] V. Iosifidis, E. Ntoutsi, Adafair: Cumulative fairness adaptive boosting, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019.
- [7] X. Song, Z. Liu, B. Jiang, Adaptive boosting with fairness-aware reweighting technique for fair classification, Expert Systems With Applications (2024).
- [8] L. B. Zhang, Brian Hu, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018.
- [9] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, K. Walker, Fairlearn: A toolkit for assessing and improving fairness in AI, Technical Report MSR-TR-2020-32, Microsoft, 2020. URL: https://www.microsoft.com/en-us/research/publication/ fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/.
- [10] S. Verma, J. Rubin, Fairness definitions explained, in: Proceedings of the International Workshop on Software Fairness, 2018, p. 1–7.