OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Fair and Interpretable Decision Rules for Binary Classification

Connor Lawless CAL379@CORNELL.EDU
Oktay Günlük ONG5@CORNELL.EDU
Cornell University

Abstract

In this paper we consider the problem of building Boolean rule sets in disjunctive normal form (DNF), an interpretable model for binary classification, subject to fairness constraints. We formulate the problem as an integer program that maximizes classification accuracy with explicit constraints on equality of opportunity. A column generation framework, with a novel formulation, is used to efficiently search over exponentially many possible rules, eliminating the need for heuristic rule mining. Compared to other interpretable machine learning algorithms, our method produces interpretable classifiers that have superior performance with respect to the fairness metric.

1. Introduction

With the explosion of artificial intelligence in recent years, automated decision making has begun taking over key decision making tasks in a variety of areas ranging from finance to driving. However, with machine learning dictating decisions as important as lending, hiring, and college admissions, a natural question is whether these algorithms are fair to all those affected. Recent results have shown machine learning algorithms to be racially biased in a range of applications ranging from facial identification in picture tagging to predicting criminal recidivism [23]. Further complicating the problem is the need for model interpretability in many applications where machine learning models complement human decision making, such as criminal justice and medicine. In this paper we aim to address these concerns, introducing a novel algorithm to build binary classifiers that are both fair and interpretable through the use of integer programming.

We focus on a well-studied interpretable class of rule sets, Boolean rules in disjunctive normal form (DNF, 'OR-of-ANDs'). For example, a DNF rule set with two rules for predicting recidivism could be $\big[$(Priors$\geq$3) and (Age$\leq$45) and (Score Factor $=$ TRUE)$\big]$ OR $\big[$(Priors$\geq$20) and (Age$\geq$45)$\big]$ where Priors, Age, and Score Factor are features related to the defendant. The fewer the rules or conditions in each rule, the more interpretable the rule set. In contrast to decision trees [2, 7, 8, 18, 24], and decision lists [3, 20, 22, 25–27], other interpretable classes of rule sets, the rules within a DNF rule-set are unordered and have been shown in a user study to require less effort to understand [21]. To build fair DNF rule sets, we start with the model in [12] which frames the problem as a large integer program (IP), generating candidate rules using a column generation (CG) framework. We keep the objective of the IP the same and add explicit constraints on fairness to control the level of acceptable "unfairness" among different subgroups. We also add additional constraints to the model as the objective function does not guarantee correctness in the presence of fairness constraints. Our approach also differs from [12] in the way we solve the pricing problem as we use a very compact formulation to generate candidate rules which reduces the computational effort significantly.

2. Fairness Metrics

We start by defining the standard supervised binary classification problem, where given a training set of n samples $(\mathbf{X}_i, y_i)$ with labels $y_i \in \{0,1\}$ and features $X_i \in \{0,1\}^p$ for $i \in I = \{1,\dots,n\}$, the goal is to generate a decision rule $d : \{0,1\}^p \to \{0,1\}$ that minimizes the expected error $\mathbb{P}(d(X) \neq Y)$ between the predicted label and the true label for unseen data. As we discuss later, assuming the data to be binary-valued is not restrictive as numeric features can be *binarized* using a sequence of thresholds and the same can be done for categorical features using one-hot encoding.

Now consider the case when each data-point also has an associated group (or protected feature) $g_i \in \mathcal{G}$ where $\mathcal{G}$ is a given discrete set. Quantifying fairness is not a straight forward task in this context and a number of metrics have been proposed in the fair machine learning literature [1, 1, 9, 14, 15, 17, 17, 19, 29]. We focus on Equality of Opportunity, a fairness criterion that requires the false negative rate to be equal across groups by enforcing the following condition [17]:

$$\mathbb{P}(d(X)=0|Y=1,G=g) = \mathbb{P}(d(X)=0|Y=1,G=g') \quad \forall g,g' \in \mathcal{G} \tag{1}$$

This metric is particularly relevant when there is a much larger societal cost to false negatives than false positives, for example in applications such as loan approval or hiring decisions, see [11, 28]. We also provide an alternative formulation of this paper based on a related notion of fairness called equalized odds in the supplementary materials (SM).

In a practical setting, it is unrealistic to expect to find classifiers that can satisfy the above criteria exactly and therefore one needs to consider how much these conditions are violated as a measure of fairness. For example, in the context of equality of opportunity, the maximum disparity among groups can be used to measure the *unfairness* of a given classifier d as follows:

$$\Delta(d) = \max_{g,g' \in \mathcal{G}} \Big| \mathbb{P}(d(X)=0|Y=1,G=g) - \mathbb{P}(d(X)=0|Y=1,G=g') \Big|. \tag{2}$$

When training the classifier d, one can then use $\Delta(d)$ in the objective function as a penalty term or can explicitly require a constraint of the form $\Delta(d) \leq \epsilon$ to be satisfied by d. We focus on the latter case as it allows for explicit control over tolerable unfairness.

3. Classification Framework: Boolean Decision Rule Sets

We now introduce our method to construct optimal DNF rule-sets for binary classification subject to fairness constraints. Note that when the input data is binary-valued, a DNF-rule set simply corresponds to checking whether a subset of features satisfies a specific combination of 0s and 1s. By ensuring that the data also includes the complement of every feature, a DNF-rule set simply checks if a subset of features are all 1 for a given data point. Consequently, if there are p binary features there can only be a finite number of $(2^p - 2)$ possible decision rules. Therefore, in theory, it is possible to enumerate all possible rules and then formulate a large integer program (IP) to select a small subset of them to minimize error on the training data under explicit fairness constraints. However from a practical perspective, it is clearly not possible to solve an exponential size IP, so instead we solve the continuous relaxation (LP) of the IP using column generation. Consequently, instead of enumerating all possible rules, we only enumerate those that can potentially improve classification error. Similar to [12] the objective of the IP is to minimize *Hamming loss*, a proxy for classification error that counts the number of rules that needs to be changed to classify a point correctly.

3.1. Integer Program Formulation

Let $\mathcal{K}$ denote the set of all possible DNF rules and $\mathcal{K}_i \subset \mathcal{K}$ be the set of rules met by data point $i \in I$. Let c_k denote the complexity of rule $k \in \mathcal{K}$ which is defined as a fixed cost of 1 plus the number of conditions in the rule. Assume that the data points are partitioned into two sets based on their labels: $\mathcal{P} = \{i \in I : y_i = 1\}$, and $\mathcal{Z} = \{i \in I : y_i = 0\}$. Additionally, for each group $g \in \mathcal{G}$ we denote the data points that have the protected feature g with $\mathcal{G}_g = \{i \in I : g_i = g\}$ and let $\mathcal{P}_g = \mathcal{P} \cap \mathcal{G}_g$ and $\mathcal{Z}_g = \mathcal{Z} \cap \mathcal{G}_g$. For simplicity, we describe the constraints assuming $\mathcal{G} = \{1, 2\}$ and note that extending it to multiple groups is straightforward. Let $w_k \in \{0, 1\}$ be a variable indicating if rule $k \in \mathcal{K}$ is selected; $\zeta_i \in \{0, 1\}$ be a variable indicating if data point $i \in \mathcal{P}$ is misclassified; and C be a parameter denoting the maximum complexity allowed. With this notation in mind, the problem of identifying the optimal rule set subject to constraints on *equality of opportunity* becomes:

$$z_{mip} = \mathbf{min} \sum_{i \in \mathcal{P}} \zeta_i + \sum_{i \in \mathcal{Z}} \sum_{k \in \mathcal{K}_i} w_k \tag{3}$$

$$\textbf{s.t.} \quad \zeta_i + \sum_{k \in \mathcal{K}_i} w_k \geq 1 \qquad i \in \mathcal{P} \tag{4}$$

$$C\zeta_i + \sum_{k \in \mathcal{K}_i} 2w_k \leq C \qquad i \in \mathcal{P} \tag{5}$$

$$\sum_{k \in \mathcal{K}} c_k w_k \leq C \tag{6}$$

$$w \in \{0,1\}^{|\mathcal{K}|}, \zeta \in \{0,1\}^{|\mathcal{P}|} \tag{7}$$

$$\frac{1}{|\mathcal{P}_1|} \sum_{i \in \mathcal{P}_1} \zeta_i - \frac{1}{|\mathcal{P}_2|} \sum_{i \in \mathcal{P}_2} \zeta_i \leq \epsilon_1 \tag{8}$$

$$\frac{1}{|\mathcal{P}_2|} \sum_{i \in \mathcal{P}_2} \zeta_i - \frac{1}{|\mathcal{P}_1|} \sum_{i \in \mathcal{P}_1} \zeta_i \leq \epsilon_1 \tag{9}$$

We denote this integer program the Master Integer Program (MIP), and it's associated linear relaxation the Master LP (MLP) (obtained by dropping the integrality constraint). Any feasible solution $(\bar{w}, \bar{\zeta})$ to (4)-(7) corresponds to a rule set $S = \{k \in \mathcal{K} : \bar{w}_k = 1\}$. Note that the objective is the Hamming loss where the first term counts the number of misclassified data-points ζ_i for $i \in \mathcal{P}$, whereas the second term adds up the total number of selected rules satisfied by data-points i for each $i \in \mathcal{Z}$. Constraint (4) identifies false negatives by forcing ζ_i to take value 1 if no rule that is satisfied by the point $i \in \mathcal{P}$ is selected. Similarly, constraint (5) ensures that ζ_i can only take a value of 1 if no rules satisfied by $i \in \mathcal{P}$ are selected. Here we use the fact that $c_k \geq 2$ for all $k \in \mathcal{K}$. Constraint (6) provides the bound on complexity of the final rule set. Finally, constraints (8) and (9) bound the maximum allowed unfairness, denoted by Δ in (2) by a specified constant $\epsilon_1 \geq 0$. If ϵ_1 is chosen to be 0, then the fairness constraint is imposed strictly. Depending on the application, ϵ_1 can also be larger than 0, in which case a prescribed level of unfairness is tolerated.

3.2. Column Generation Framework

To solve the LP relaxation of the MIP, called the MLP, using the column generation framework [10], we start with a small subset $\hat{\mathcal{K}} \subset \mathcal{K}$ of all possible rules and solve an LP restricted to the variables associated with these rules only. Once this small LP is solved, we use its optimal dual solution to identify a missing variable (rule) that has a negative reduced cost [5]. The search for such a variable is called the *pricing problem* and, in our case, this can be done by solving a separate integer program. If a variable with a negative reduced cost is found, then $\hat{\mathcal{K}}$ is augmented with the associated rule and the LP is solved again and the this process is repeated until no such variables can be found.

Given a possibly empty subset of rules $\hat{\mathcal{K}} \subset \mathcal{K}$, let the restricted MLP, defined by (3)-(6), (8)-(9) and denoted by RMLP, be the restriction of MLP to the rules in $\hat{\mathcal{K}}$. Let $(\mu, \alpha, \lambda, \gamma^1, \gamma^2)$ be an optimal *dual* solution to RMLP, where variables $\mu, \alpha, \lambda \geq 0$ are associated with constraints (4), (5), and (6), respectively. Variables γ^1 and γ^2 are associated with fairness constraints (8) and (9). We now formulate an integer program to find a $k \in \mathcal{K}$ with the minimum reduced cost $\hat{\rho}_k$. Remember that a decision rule corresponds to a subset of the binary features J and classifies a data point with a positive response if the point has all the features selected by the rule. Let variable $z_j \in \{0,1\}$ for $j \in J$ denote if the rule has feature j and let variable $\delta_i \in \{0,1\}$ for $i \in I$ denote if the rule misclassifies sample i. Using these variables, the complexity of a rule can be computed as $(1 + \sum_{j \in J} z_j)$. We now construct the full pricing problem with the reduced cost in the objective:

$$z_{cg} = \textbf{min} \quad \sum_{i \in \mathcal{Z}} \delta_i + \sum_{i \in \mathcal{P}} (2\alpha_i - \mu_i)\delta_i + \lambda(1 + \sum_{j \in J} z_j) \tag{10}$$

$$\textbf{s.t.} \quad D\delta_i + \sum_{j \in S_i} z_j \leq D \qquad i \in I^- \tag{11}$$

$$\delta_i + \sum_{j \in S_i} z_j \geq 1 \qquad i \in I^+ \tag{12}$$

$$\sum_{j \in J} z_j \leq D \tag{13}$$

$$z \in \{0,1\}^{|J|}, \delta \in \{0,1\}^{|\mathcal{P}|} \tag{14}$$

where the set $I^- \subseteq I$ contains the indices of δ_i variables that have a negative coefficient in the objective, and $I^+ = I \setminus I^-$. The objective is the reduced cost for the generated rule. Note that variable w_k does not appear in constraints (8) or (9) in RMLP and consequently the objective does not involve variables γ^1 or γ^2. Also note that constraints (11) and (12) to ensure that δ_i accurately reflects whether the new rule classifies data point i with a positive label, and constraint (13) puts an explicit bound on the complexity of any rule using the parameter D. This individual rule complexity constraint can be set independently of C in the master problem or simply be set to $C - 1$.

4. Experiments

To benchmark the performance of our algorithm, we tested it on three fair machine learning datasets: *default* [13], *adult* [13], and *compas* [4]. Data processing and implementation details are included the supplementary materials (SM). Figure 1 (a) shows the trade-off between the fairness constraint for equality of opportunity when training and the realized false negative rate. As we relax the fairness constraint both the realized train and test set fairness decreases (i.e. the gap between groups grows). This disproportionately impacts the false negative rate for the first group, underscoring the importance of finding fair classifiers. Figure 1 (b) shows that increasing rule set complexity leads to lower false negative rates across groups, underscoring the inherent trade-off in interpretability and fairness.

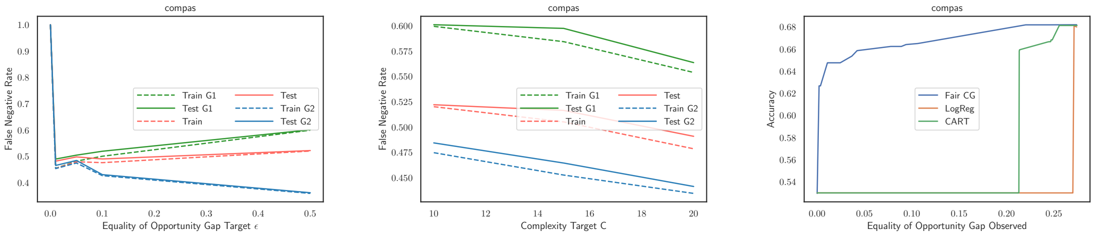

Figure 1: Impact of the equality of opportunity fairness constraint (a) and complexity constraint (b) on false negative rate for the *compas* dataset. (c) Performance of FairCG benchmarked against other interpretable models on *compas* dataset. For all plots, if group is unspecified line is for all data.

We also compared the performance of our algorithm with two other interpretable binary classification models, CART and Logistic Regression. For all three models we varied the hyper-parameters, performing 10-fold cross validation for each parameter, to generate the accuracy fairness trade-offs. Figure 1 (c) plots the efficient frontier for accuracy-fairness for CART, Logistic Regression as well as our own algorithm for the *compas* dataset. Note that our algorithm can deal with the accuracy/fairness trade-off much better. Table 1 summarizes each algorithm's performance when tuned for accuracy and fairness separately. For each dataset we report the standard deviation in parenthesis. While CART is able to achieve superior predictive accuracy on some datasets, our algorithm is able to achieve comparable accuracy under much stricter fairness constraints. Overall this shows that our framework is able to build interpretable models that have competitive accuracy and substantially improved fairness. Computational results for other datasets can be found in the SM.

Table 1: Mean Accuracy and Fairness Results for Equality of Opportunity

		Adult		Compas		Default	
		Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
Fair CG	Tuned for Acc	82.9 (0.2)	9.4 (0.4)	68.2 (1.2)	24.5 (5.3)	82.0 (0.6)	0.5 (1.2)
	Tuned for Fair	78.4 (0.4)	0.3 (0.3)	53.0 (1.6)	0 (0)	77.9 (0.4)	0 (0)
CART	Tuned for Acc	85.5 (0.3)	15.9 (5.1)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Tuned for Fair	85.4 (0.5)	8.4 (4.3)	65.8 (2.3)	21.3 (6.1)	82.0 (1.4)	2.5 (1.9)
LR	Tuned for Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Tuned for Fair	79.8 (0.6)	3.6 (3.2)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)

5. Conclusion

In this work we introduce an algorithm to build DNF rule sets for binary classification that are both interpretable and fair using an integer programming formulation. While preliminary empirical results show that our algorithm is able to achieve competitive accuracy and superior fairness on standard fair machine learning data-sets, it remains future work to benchmark our algorithm against other fair interpretable models [2, 6, 18] when those models have been optimized for the same fairness metrics.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification, 2018.

[2] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making, 2019.

[3] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 35–44, 2017.

[4] Julia Angwin, Jeff Larson, Surya Mattu, and Laura Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. Available at https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (2016).

[5] Mokhtar S. Bazaraa, John Jarvis, and Hanif D. Sherali. *Linear programming and network flows*. Wiley, 2010.

[6] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression, 2017.

[7] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Mach. Learn.*, 106(7):1039–1082, July 2017.

[8] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[9] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, 21:277–292, 09 2010. doi: 10.1007/s10618-010-0190-x.

[10] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. *Integer programming*. Springer, 2014.

[11] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning, 2018.

[12] Sanjeeb Dash, Oktay Günlük, and Dennis Wei. Boolean decision rules via column generation, 2018.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness through awareness, 2011.

[15] Harrison Edwards and Amos Storkey. Censoring representations with an adversary, 2015.

[16] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. URL http://www.gurobi.com.

[17] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning, 2016.

[18] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, page 869–874, USA, 2010. IEEE Computer Society. ISBN 9780769542560. doi: 10.1109/ICDM.2010.50. URL https://doi.org/10.1109/ICDM.2010.50.

[19] Faisal Kamiran, Indrė Žliobaitė, and Toon Calders. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowledge and Information Systems*, 1:in press, 06 2012. doi: 10.1007/s10115-012-0584-8.

[20] Himabindu Lakkaraju and Cynthia Rudin. Learning cost-effective and interpretable treatment regimes. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 166–175, Fort Lauderdale, FL, USA, 20–22 Apr 2017.

[21] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1675–1684, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939874. URL https://doi.org/10.1145/2939672.2939874.

[22] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.*, September(3):1350–1371, 09 2015.

[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning, 2019.

[24] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[25] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[26] Tong Wang and Cynthia Rudin. Learning Optimized Or's of And's, November 2015. arXiv:1511.02210.

[27] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable Bayesian rule lists. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1013–1022, 2017.

[28] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification, 2015.

[29] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact. *Proceedings of the 26th International Conference on World Wide Web - WWW '17*, 2017. doi: 10.1145/3038912.3052660. URL http://dx.doi.org/10.1145/3038912.3052660.

Appendix A. Supplementary Materials

A.1. Equalized Odds

In this section, we extend the formulation from this paper to consider another notion of fairness: equalized odds. Recall that thee equality of opportunity simply ensures the false negative rate is equal across groups.A stricter condition on the classifier is to require that the classification error is equal across all groups and for both the positive and negative classes within those groups [17]. This requirement prevents possible trade-off between false negative and false positive errors across groups and can be seen as a generalization of the equality of opportunity criterion to include false positives. To achieve equalized odds, together with equation (1), the following condition is also enforced:

$$\mathbb{P}(d(X)=1|Y=0,G=g)=\mathbb{P}(d(X)=1|Y=0,G=g') \quad \forall g,g'\in\mathcal{G} \tag{15}$$

This notion of fairness may be overly restrictive in applications where there is little difference in societal cost between loss for the two response classes.

To incorporate a notion of equalized odds into our IP formulation, we extend the notion of equalized odds to the hamming loss setting (henceforth denoted hamming equalized odds). Specifically, to bound the disparity in false positive rate we bound the disparity in the hamming loss terms for the negative class. To that end, together with constraints (8) and (9) , we include the following constraints in the formulation:

$$\frac{1}{|\mathcal{Z}_1|}\sum_{i\in\mathcal{Z}_1}\sum_{k\in\mathcal{K}_i} w_k - \frac{1}{|\mathcal{Z}_2|}\sum_{i\in\mathcal{Z}_2}\sum_{k\in\mathcal{K}_i} w_k \le \epsilon_2 \tag{16}$$

$$\frac{1}{|\mathcal{Z}_2|}\sum_{i\in\mathcal{Z}_2}\sum_{k\in\mathcal{K}_i} w_k - \frac{1}{|\mathcal{Z}_1|}\sum_{i\in\mathcal{Z}_1}\sum_{k\in\mathcal{K}_i} w_k \le \epsilon_2, \tag{17}$$

where $\epsilon_2 \ge 0$ is a given constant. While constraints (8) and (9) bound the mis-classifications disparity for the positive class $\mathcal{P}$, constraints (16) and (17) do the same for negative class $\mathcal{Z}$ using hamming loss. The tolerance parameter ϵ_2 in (16) and (17) can be set equal to ϵ_1 in (8) and (9), or, alternatively, ϵ_1 and ϵ_2 can be chosen separately. Note that we normalize the hamming loss terms to account for the difference in group sizes and positive response rates between groups.

Now we consider how to adapt the pricing problem. In this case the RMLP is defined by (3)-(6), (8)-(17) and note that unlike (8) and (9), constraints (16) and (17) do involve variables w_k. Let $(\mu,\alpha,\lambda,\gamma^1,\gamma^2,\gamma^3,\gamma^4)$ be an optimal dual solution to RMLP, where variables γ^3 and γ^4 are associated with fairness constraints (16) and (17), respectively. Using this dual solution, the reduced cost of a variable w_k associated with $k\notin\hat{\mathcal{K}}$ is similar to the expression in (**??**), except, it has the following 4 additional terms:

$$\sum_{i\in\mathcal{Z}_1}\frac{\gamma_3}{|\mathcal{Z}_1|}\mathbb{1}_{\{k\in\mathcal{K}_i\}} - \sum_{i\in\mathcal{Z}_1}\frac{\gamma_4}{|\mathcal{Z}_1|}\mathbb{1}_{\{k\in\mathcal{K}_i\}} - \sum_{i\in\mathcal{Z}_2}\frac{\gamma_3}{|\mathcal{Z}_2|}\mathbb{1}_{\{k\in\mathcal{K}_i\}} + \sum_{i\in\mathcal{Z}_2}\frac{\gamma_4}{|\mathcal{Z}_2|}\mathbb{1}_{\{k\in\mathcal{K}_i\}} \tag{18}$$

Consequently, the pricing problem becomes

$$\begin{aligned} z_{cg} \;=\; \mathbf{min}\;\; & (1+\frac{\gamma_3-\gamma_4}{|\mathcal{Z}_1|})\sum_{i\in\mathcal{Z}_1}\delta_i + (1+\frac{\gamma_4-\gamma_3}{|\mathcal{Z}_2|})\sum_{i\in\mathcal{Z}_2}\delta_i + \sum_{i\in\mathcal{P}}(2\alpha_i-\mu_i)\delta_i + \lambda(1+\sum_{j\in J} z_j) \\ \textbf{s.t.}\;\; & (11)-(14). \end{aligned}$$

A.2. Datasets and data processing

We use three common fair machine learning datasets: Adult, Compas and Default. Both Adult[1] and Default[2] can be found on the UCI machine learning dataset repository [13]. Both datasets were unchanged from the data available at the referenced links. For the adult dataset we just use the training data provided, and for default we use the the entire dataset provided. For the COMPAS data from ProPublica [4] we use the fair machine learning cleaned dataset[3]. Following the methodology of [28] we also restrict the data to only look at african american and caucasian respondents - filtering all datapoints that belong to another race and creating a new binary column race which indicates whether or not the respondent was african american. The final version of each dataset we used can be found on our github page [4]. A summary of our data, and the sensitive attributes we use for group identification, is included below.

Table 2: Overview of datasets

Dataset	Examples	Features	Sensitive Variable
Adult	32,561	14	Gender
Compas	5,278	7	Race
Default	30,000	23	Gender (X2)

To convert the data to be binary-valued, we use a standard methodology also used in [12]. For categorical variables we use one-hot encoding and include both the encoding and it's negation as features. For numerical variables we compare values against the sample deciles for that column and include both the comparison and it's negation (i.e. $X_j \leq 1, X_j \leq 2$ and $X_j > 1, X_j > 2$).

A.3. Implementaion Details

We use the Python interface of Gurobi [16] to solve the linear and integer programs. To solve the MLP we use a barrier interior point method with the default crossover parameter. For the pricing problem we use the default settings, and return all solutions generated during the algorithm's run with negative reduced costs. For the MIP we also return all solutions generated and use the rule set that has the highest classification accuracy on the training data.

In many problems, solving either the integer pricing problem or the MLP to optimality can be computationally intensive. To place a practical limit on the overall problem, we set time limits on the pricing problem and the overall column generation (CG) process to solve the MLP. We employ a standard time limit argument in Gurobi. Once the MLP is solved to (near) optimality, if the current formulation has more than 1000 rules, we select 1000 rules with the lowest reduced cost and solve the MIP using these rules only. We also set a time limit for solving the MIP and return all feasible solutions found by the time the limit is reached.

For small problems we employ the CG framework described in Section 3.2, however for larger problems we use an approximate version of the framework to limit the overall run time. For large datasets, the integer programming formulation for the pricing problem may become computationally

[1] https://archive.ics.uci.edu/ml/datasets/adult
[2] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[3] https://www.kaggle.com/danofer/compass
[4] excluded for double blind review

intractable and therefore we solve the pricing problem for a sub-sample selected uniformly at random of the full dataset. The sub-sampled problem on average has 2000 rows and 100000 non-zeros. In addition to solving the pricing problem, we also use a greedy heuristic (similar to [12]) for generating rules with up to five features. We start with all rules with a single feature and compute their reduced costs using the dual solution of the RMLP. We keep the best 20 rules and then iteratively build up rule size by testing all k-feature rules built off of the 20 retained (k-1)-feature rules. All rules with negative reduced costs (even those not in the best 20 for each rule size) are returned. In practice, we found employing the heuristic for a fixed number of CG cycles and then switching to the IP formulation to 'fine-tune' the rule-set produced the best results. If a large number of candidate rules are generated (for both the heuristic and integer programming formulation) we return the 100 rules with the lowest reduced costs.

We used ten-fold cross validation and performed a two-stage process to generate candidate rules and build a rule-set. First, we run the CG algorithm on the training data for a subset of potential hyper-parameters (typically 3 different complexity limits and epsilon values). Then, we use these candidate rules and solve the master IP for a grid of potential epsilon values and complexity limits. To select which complexity values to test we look at which complexity value leads to the best cross-validated accuracy in the problem without fairness constraints and test values around and including it. We use a five minute time limit for the solving the MLP and MIP, and a 45 second time limit for each iteration of the pricing problem.

A.4. Experimental Set-up

Our experiments were run on a microsoft azure data science virtual machine using Standard DS3 v2 (4 vcpus, 14 GiB memory) hardware [5]. Our python environment was configured with anaconda and included the following package dependencies:

Table 3: Overview of package dependencies

Package	Version
Python	3.7.6
Gurobi	9.0.1
Numpy	1.18.1
Pandas	1.0.0

For our experiments we took a two-phase approach. During the first rule generation phase, we ran our column generation algorithm with a set of different hyper-parameters to generate a set of potential rules. We then solved the master IP with the set of candidate rules and a larger set of hyperparameters to generate the curves included in the body of the report and these supplementary materials. For all datasets we used $\{0.2, 1\}$ for the first rule generatioin phase for the fairness constraint ϵ, and $\{0, 0.01, 0.03, 0.05, 0.1, 0.2, 0.5, 1\}$ for the evaluation fairness ϵ. For hamming equalized odds, we also set $\epsilon_1 = \epsilon_2$ and use the same training and test epsilons. For the rule complexity parameter C we use the following values for the first phase:

- **Adult**: $\{60, 80, 110\}$

[5] https://azuremarketplace.microsoft.com/en-ca/marketplace/apps/microsoft-dsvm.dsvm-win-2019?tab=Overview

- **Compas**: $\{5, 15, 30\}$
- **Default**: $\{5, 15, 30\}$

And the following values for the second phase:

- **Adult**: $\{80, 90, 100\}$
- **Compas**: $\{10, 15, 20\}$
- **Default**: $\{10, 15, 20\}$

Appendix B. Additional Results

This section includes a comprehensive look at our results across both fairness criterion and all three data-sets. We break down the results first by metric, and then by data-set. We conclude by including a section detailing the results of our algorithm benchmarked against CART and logistic regression.

B.1. Equality of Opportunity

For each of our three datasets we present two sets of plots. The first shows the impact of changing the ϵ in fairness constraints 8 and 9 on the false negative rate overall and for each group, the gap in false negative rate between both groups, and the predictive accuracy. For these plots, we vary ϵ and select the complexity that leads to the best results for that ϵ on the training data (i.e. highest accuracy, lowest false negative rate). The second set of plots show the impact of changing the C in constraint 6 on accuracy and false negative rate. Similar to the first set of plots, we vary the complexity parameter and select the ϵ fairness constraint that leads to the best results, in practice this always translates to setting $\epsilon = 1$ (i.e. no fairness constraint). Throughout we use dashed lines to denote the performance of our algorithm on the training data and solid lines for test data. We use G1 and G2 to denote the two groups defined by the sensitive variable.

B.1.1. ADULT

The first plot in Figure 2 shows that as we relax the fairness constraint we first see a large decrease in the false negative rate (FNR) and then a gradual decrease in the FNR for G1 and increase for G2. The high FNR at $\epsilon = 0$ is simply caused by the fact that the algorithm cannot find a non-degenerate (i.e. non-empty) rule set with perfect fairness, and by default classifies everything as the negative class. Thus relaxing the constraint even slightly leads to a non-degenerate rule set that achieves lower FNR. As we relax it even more, the algorithm is able to get better overall loss by classifying the majority group (G1) more accurately at the expense of the minority group (G2). This underscores the importance of fair machine learning, as better accuracy comes at a disproportionate cost to one group. The second plot shows that the realized train and test equality of opportunity gap (i.e. difference in FNR rates between groups) align well with the imposed constraint. Finally we see that there is overall a small change in predictive accuracy as we relax the fairness constraint, demonstrating the ability of our algorithm to find rule sets with equivalent accuracy that are able to trade off false positive and false negatives.

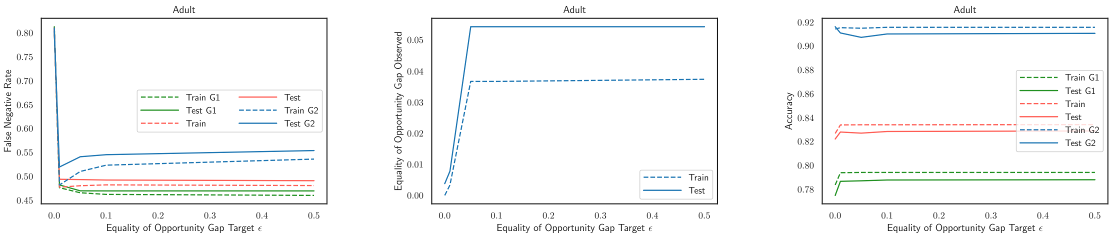

Figure 2: Impact of the fairness constraint on fairness and accuracy for the adult dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

The second set of plots show that changing complexity had a relatively small impact on predictive accuracy. However, we are able to achieve models with lower false negative rates as we increase complexity (and thus decrease interpretability). This highlights the inherent trade-off in interpretability and fairness.

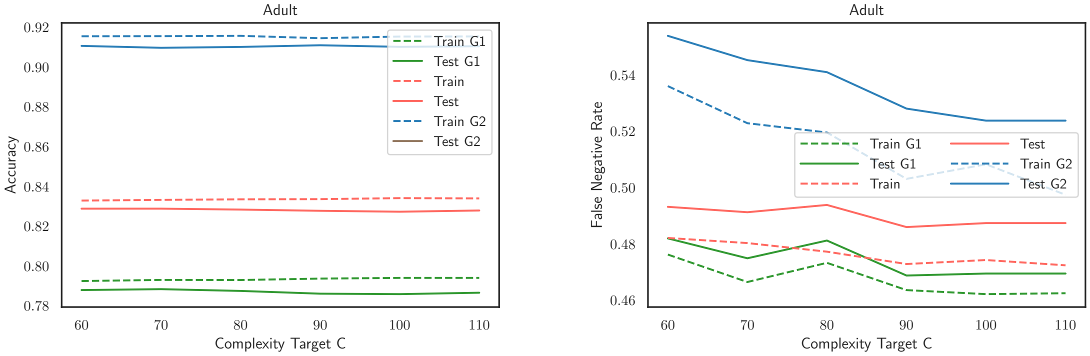

Figure 3: Impact of rule set complexity constraint on fairness and accuracy for the adult dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

B.1.2. COMPAS

The results on the compas data-set show similar trends as the adult dataset. Of note, we see an even tighter accordance between fairness on the train and test sets and a starker impact of relaxing the fairness constraint on predictive accuracy. This could indicate that the predictive task in compas is much more sensitive to fairness (i.e. fair classifiers come at a higher cost in accuracy) than the adult dataset.

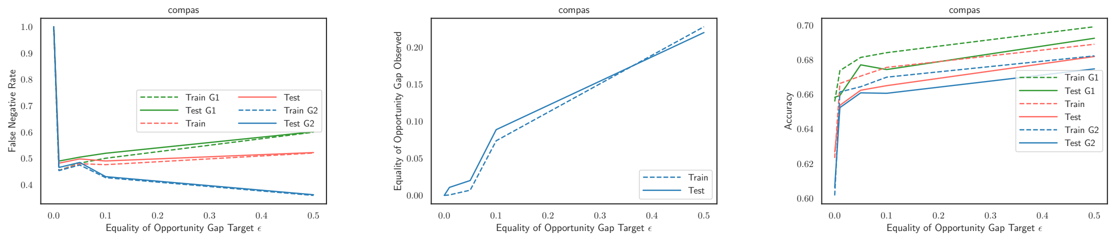

Figure 4: Impact of the fairness constraint on fairness and accuracy for the compas dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

We also see similar trends in the relationship between complexity and accuracy and fairness to the adult data-set. The small exception being the larger impact of complexity on predictive accuracy.

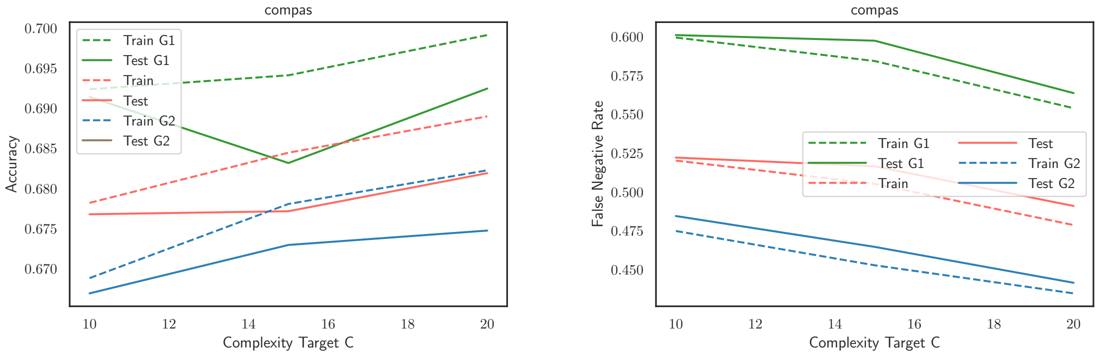

Figure 5: Impact of rule set complexity constraint on fairness and accuracy for the compas dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

B.1.3. Default

In the default data-set, our algorithm sees practically no impact on accuracy or FNR after relaxing the fairness constraint past 0.01. This indicates that the default dataset is an easier problem from a fairness perspective (i.e. we can get equivalent predictive performance with very tight adherence to fairness criteria).

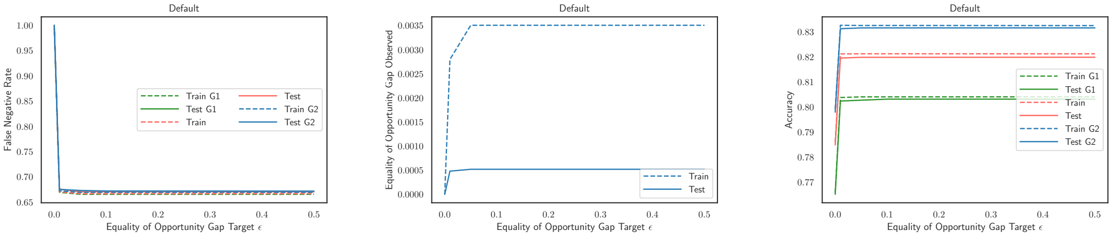

Figure 6: Impact of the fairness constraint on fairness and accuracy for the default dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

We see the same trends in complexity and accuracy/fairness as the adult dataset in the default dataset.

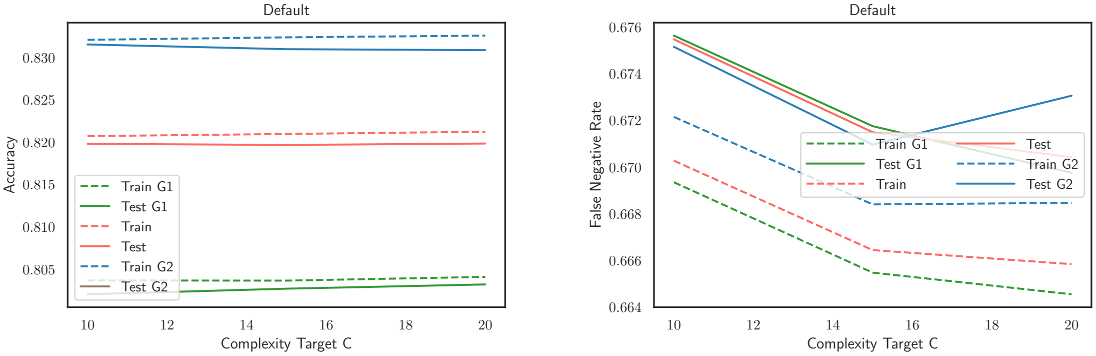

Figure 7: Impact of rule set complexity constraint on fairness and accuracy for the default dataset and the equality of opportunity metric, where G1 and G2 represent the results for each group separately.

B.2. Hamming Equalized Odds

Similar to the equality of opportunity metric, for each of our three datasets we present two sets of plots. The first shows the impact of changing the ϵ in fairness constraints 8, 9,16 and 17. For our experiments we set $\epsilon_1 = \epsilon_2 = \epsilon$ (i.e. use the same fairness tolerance for both sets of constraints). When plotting our results, we break down the overall hamming loss into two terms: the (+) term that represents loss coming from data points with the label 1, and a (-) term that represents loss coming from data points with the label 0. The (+) term is equivalent to the false negative rate. Our first set of plots explorers the impact of changing ϵ on both the (+) and (-) hamming loss terms as well as their gap (i.e. the different in each loss term between the two groups). For brevity, we refer to hamming equalized odds as HEO. The second set of plots explores the relationship between the complexity parameter C and both the hamming loss terms observed and overall accuracy. We follow the same procedure as the equality of opportunity metric, changing ϵ and C for each set of plots respectively, and choosing the other hyper parameters that maximize results for each ϵ and C separately.

B.2.1. ADULT

As we relax the fairness constraint we see a decrease in the positive term of the hamming loss and and increase in the negative term. This is because our algorithm by default classifies all data points as being part of the negative class, so as we relax the fairness constraint and our algorithm finds non-degenerate rule sets the FNR decreases and the negative hamming loss term increases. Similar to the equality of opportunity metric, we see relaxing the fairness constraint disproportionately favors G1 for the positive term. We also see a larger increase in accuracy from relaxing the fairness constraint compared to the equality of opportunity metric. This falls from the fact that our HEO formulation adds additional constraints to the equality of opportunity formulation.

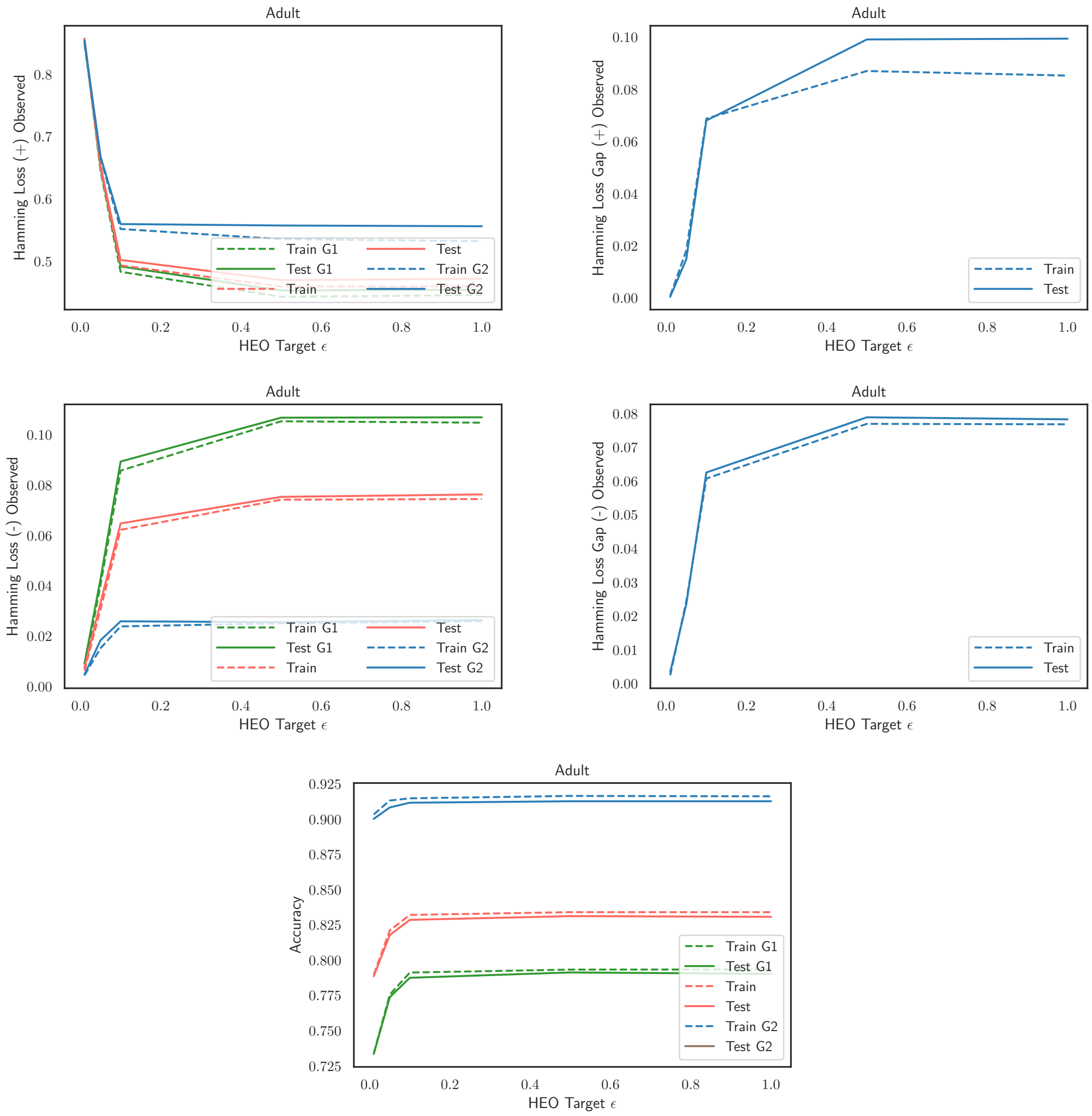

Figure 8: Impact of the fairness constraint on fairness and accuracy for the Adult dataset and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

We see a similar impact of complexity on accuracy as the previous fairness metric. As complexity increases, we see a small to negligible increase in accuracy. However, unlike the equality of opportunity metric we now have two different views on the relationship of complexity and fairness. The positive hamming loss terms follows the same result as the equality of opportunity metric, but we see an increase in the negative hamming loss term. The cause of this seemingly contradictory increase is that as we increase complexity we find non-degenerate rule sets that by nature of not categorizing all points negative by default lead to higher hamming loss terms for the negative class.

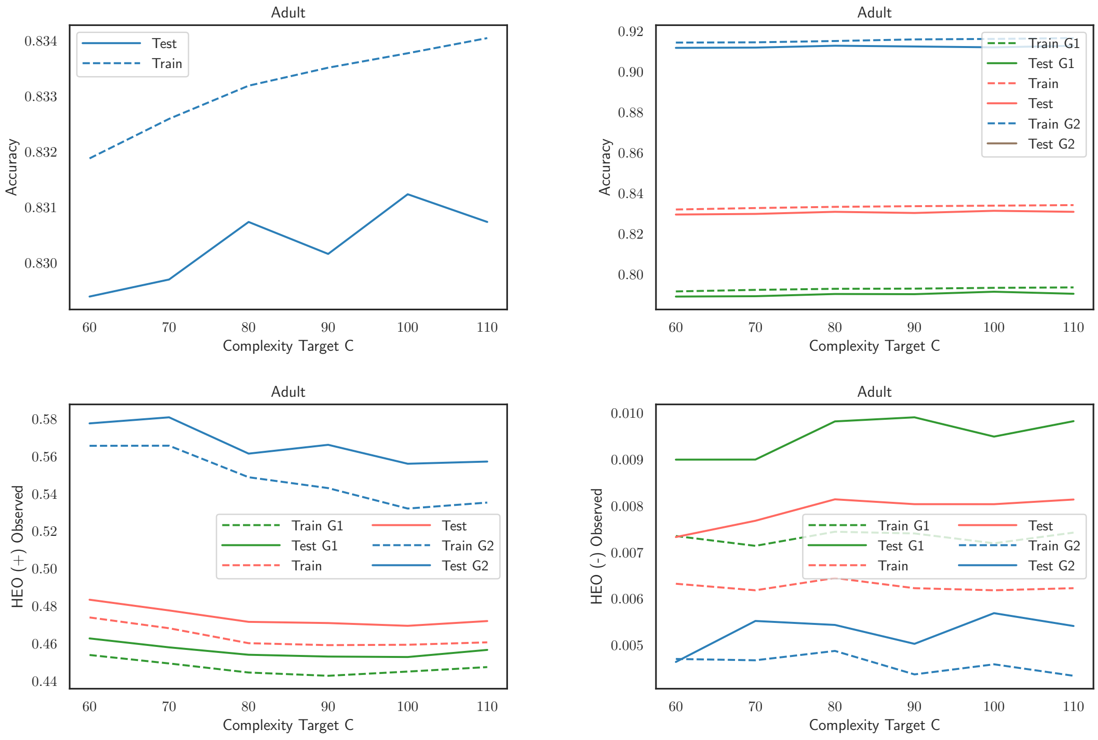

Figure 9: Impact of rule set complexity constraint on fairness and accuracy for the Adult dataset and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

B.2.2. Compas

The compas plots show a similar trend to the adult dataset, and the compas results with the other fairness criterion. Likewise, we see a similar relationship between complexity and both accuracy and fairness.

B.2.3. Default

Our results on the default dataset follow the same trends as the other datasets for both the fairness and complexity plots.

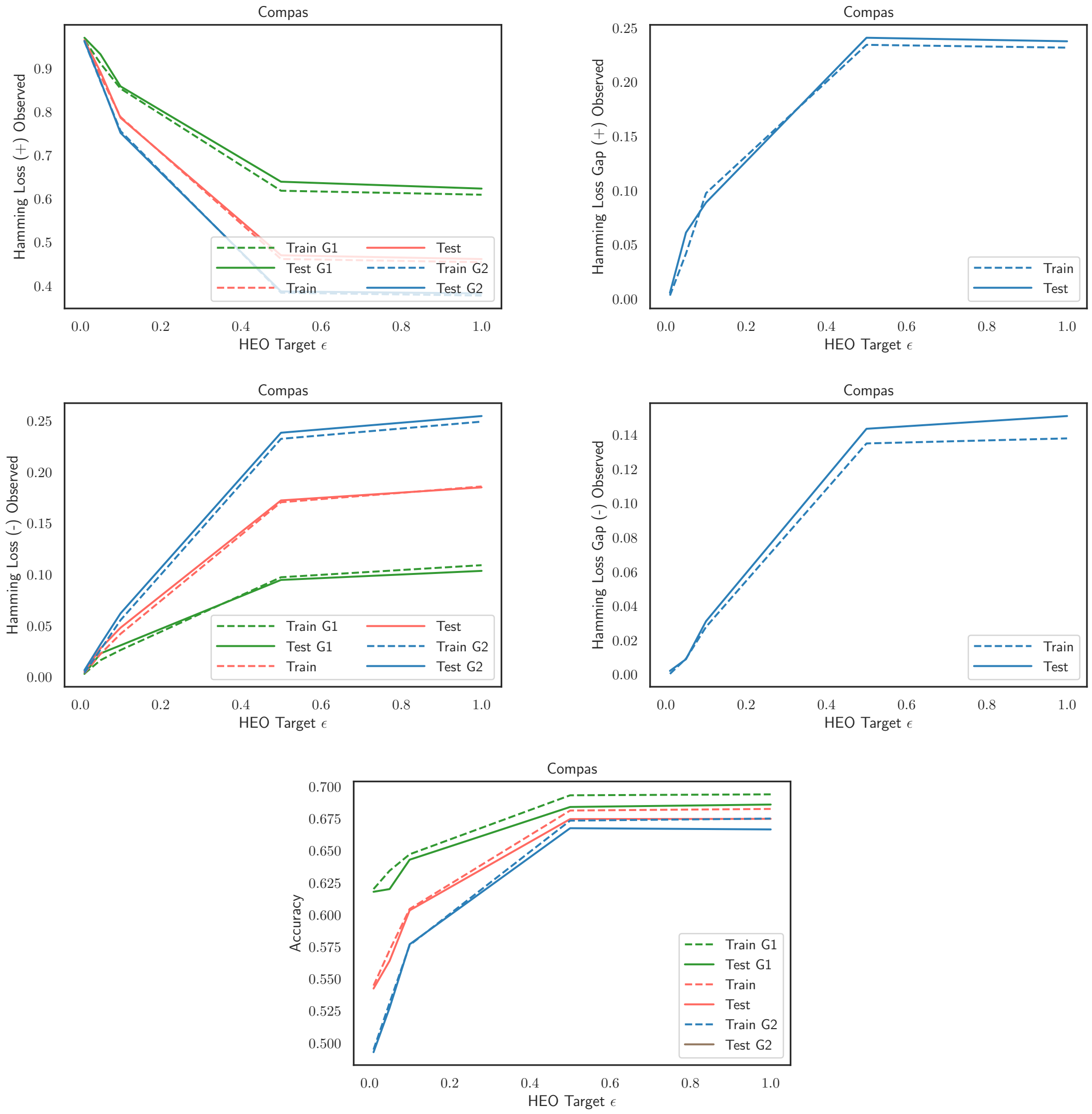

Figure 10: Impact of the fairness constraint on fairness and accuracy for the Compas data-set and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

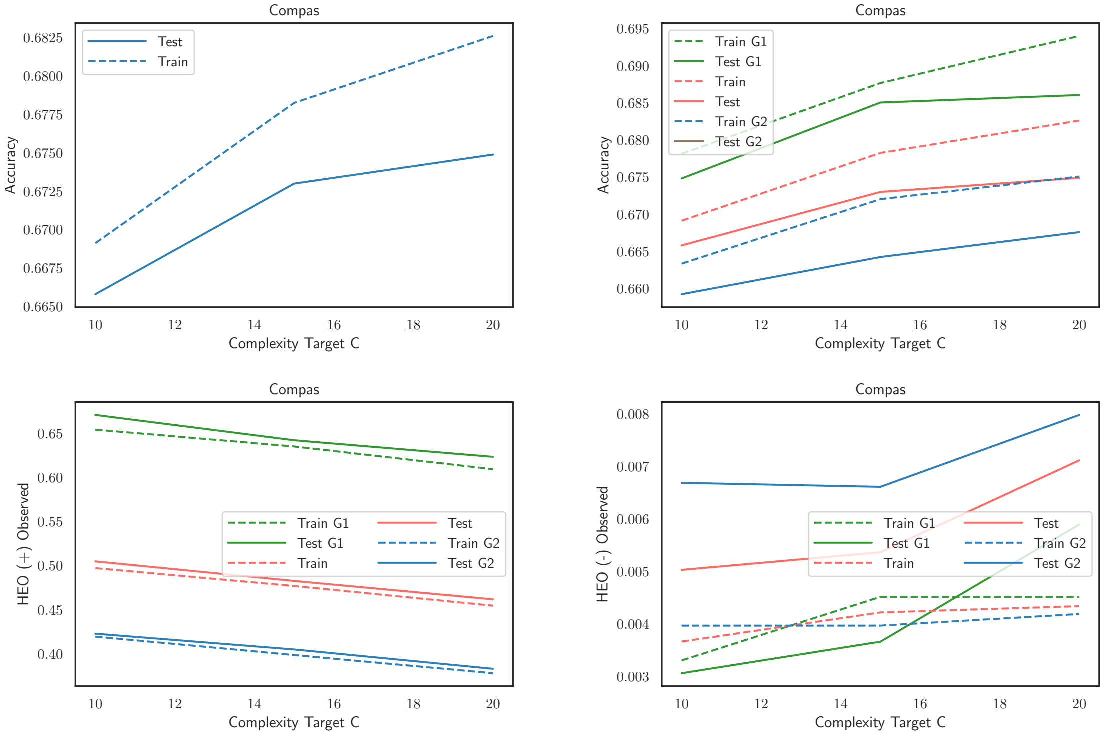

Figure 11: Impact of rule set complexity constraint on fairness and accuracy for the Compas dataset and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

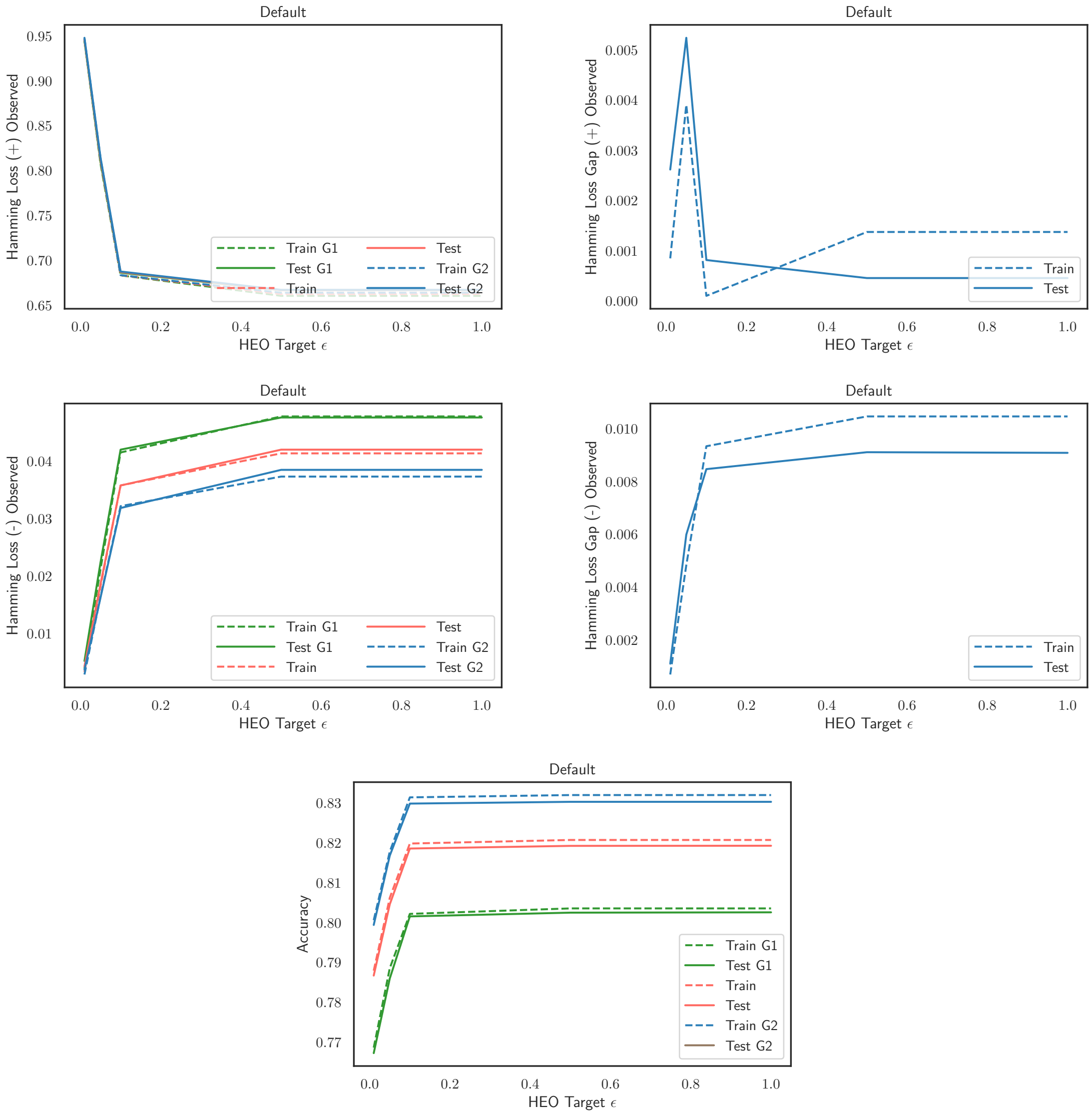

Figure 12: Impact of the fairness constraint on fairness and accuracy for the Default data-set and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

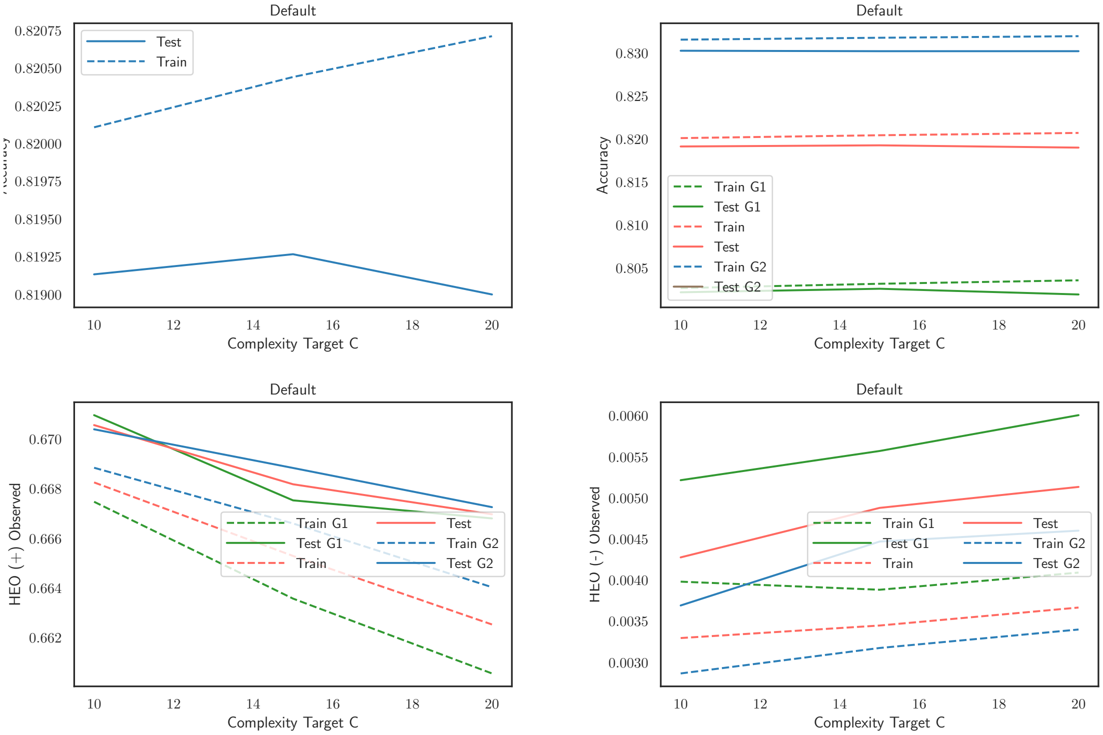

Figure 13: Impact of rule set complexity constraint on fairness and accuracy for the Default dataset and the hamming equalized odds criterion, where G1 and G2 represent the results for each group separately.

B.3. Benchmarking Results

Our final set of results compares the performance of Fair CG to two interpretable machine learning models: decision trees and logistic regression (both using the scikit-learn implementations). For these experiments we varied the hyper-parameters of both decision trees and logistic regression and performed 10-fold cross validation to generate the achievable test-set fairness and accuracy values. For decision trees we changed the min samples per leaf parameter, and logistic regression we changed the regularization penalty weight C. We then plot the fairness-accuracy curve for both algorithms and Fair CG. For the HEO metric we plot fairness with respect to the gap in both the positive and negative hamming loss terms. We also summarize these results in a table with the best models when hyper-parameters are selected to maximize accuracy or minimize fairness respectively. The table reports both the mean results across 10-fold cross validation with the standard deviation in brackets.

B.3.1. EQUALITY OF OPPORTUNITY

For the equality of opportunity metric, we see that our algorithm strictly dominates logistic regression and is able to achieve competitive accuracy with tighter fairness adherence than CART. However, CART is able to achieve better accuracies without any fairness constraints.

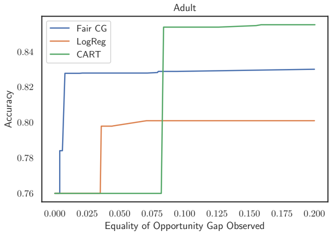

Figure 14: (a) Equality of Opportunity

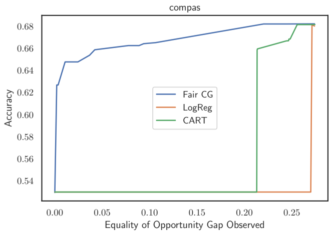

Figure 15: (b-1) Hamming Loss Gap (+ term)

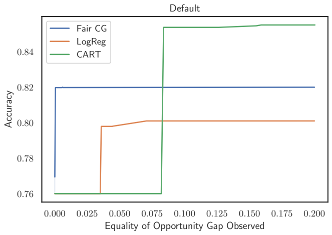

Figure 16: (b-2) Hamming Loss Gap (- term)

Figure 17: Accuracy-Fairness trade-off for our algorithm, CART and logistic regression when trained with equality of opportunity metric. Fairness is with respect to the gap in false negative rate between groups (i.e. equality of opportunity gap).

Table 4: Mean Accuracy and Fairness Results for Equality of Opportunity

		Adult		Compas		Default	
		Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
Fair CG	Best Acc	82.9 (0.2)	9.4 (0.4)	68.2 (1.2)	24.5 (5.3)	82.0 (0.6)	0.5 (1.2)
	Best Fair	78.4 (0.4)	0.3 (0.3)	53.0 (1.6)	0 (0)	77.9 (0.4)	0 (0)
CART	Best Acc	85.5 (0.3)	15.9 (5.1)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Best Fair	85.4 (0.5)	8.4 (4.3)	65.8 (2.3)	21.3 (6.1)	82.0 (1.4)	2.5 (1.9)
LR	Best Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Best Fair	79.8 (0.6)	3.6 (3.2)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)

Table 5: Mean Accuracy and Fairness Results for Hamming Equalized Odds (+ term)

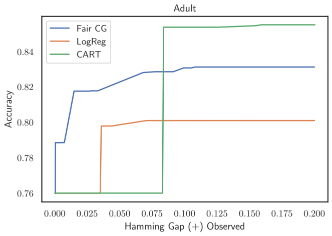

Figure 18: (a) Equality of Opportunity

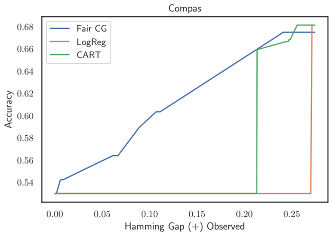

Figure 19: (b-1) Hamming Loss Gap (+ term)

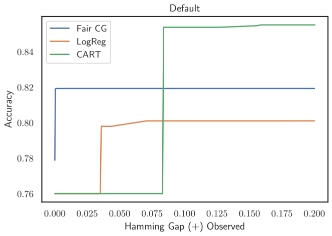

Figure 20: (b-2) Hamming Loss Gap (- term)

Figure 21: Accuracy-Fairness trade-off for our algorithm, CART and logistic regression when trained with hamming equalized odds. Fairness is with respect to the gap in hamming loss (+) between groups.

		Adult		Compas		Default	
		Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
Fair CG	Best Acc	83.1 (0.6)	10.8 (0.5)	67.5 (1.7)	24.5 (5.3)	81.9 (0.6)	0.1 (0.8)
	Best Fair	76.0 (0.5)	0 (0)	53.0 (1.7)	0 (0)	77.9 (0.5)	0 (0)
CART	Best Acc	85.5 (0.3)	15.9 (5.1)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Best Fair	85.4 (0.5)	8.4 (4.3)	65.8 (2.3)	21.3 (6.1)	82.0 (1.4)	2.5 (1.9)
LR	Best Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Best Fair	79.8 (0.6)	3.6 (3.2)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)

B.3.3. HAMMING EQUALIZED ODDS (-)

We conclude by looking at our results for fairness with respect to the negative term of hamming loss. Similar to the equality of opportunity metric, we see that our algorithm strictly dominates logistic regression and is able to achieve competitive accuracy with tighter fairness adherence than CART.

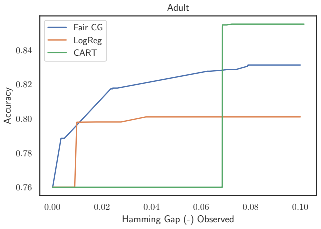

Figure 22: (a) Equality of Opportunity

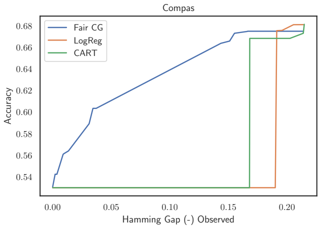

Figure 23: (b-1) Hamming Loss Gap (+ term)

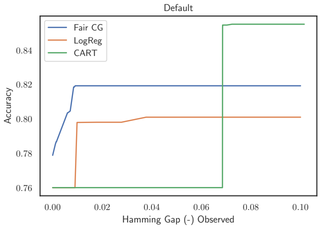

Figure 24: (b-2) Hamming Loss Gap (- term)

Figure 25: Accuracy-Fairness trade-off for our algorithm, CART and logistic regression when trained with hamming equalized odds. Fairness is with respect to the gap in hamming loss (-) between groups.

Table 6: Mean Accuracy and Fairness Results for Hamming Equalized Odds (- term)

		Adult		Compas		Default	
		Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
Fair CG	Best Acc	83.1 (0.6)	7.9 (0.4)	67.5 (1.7)	24.5 (5.3)	81.9 (0.6)	0.9 (1.1)
	Best Fair	76.0 (0.5)	0 (0)	53.0 (1.7)	0 (0)	81.9 (0.6)	0 (0)
CART	Best Acc	85.5 (0.3)	7.2 (0.5)	68.1 (1.9)	25.6 (6.2)	82.1 (1.5)	3.0 (2.7)
	Best Fair	85.3 (0.5)	6.8 (0.5)	66.8 (2.6)	16.8 (5.4)	82.0 (1.3)	1.1 (0.6)
LR	Best Acc	80.1 (1.1)	7.06 (8.0)	68.1 (1.6)	27.1 (7.6)	77.9 (1.7)	0 (0)
	Best Fair	79.8 (0.6)	1.0 (0.5)	67.5 (1.2)	19.1 (4.5)	77.9 (1.7)	0 (0)