CMSC 828Z Project Report : Rawlsian Gradient Descent

Soumik Mukhopadhyay

I. Introduction

Dr. John Rawls [1] is a political philosopher who had proposed the the ideal theory of "justice as fairness " in his book "A Theory of Justice" [2]. In this he imagines a hypothetical situation where no one knows what attributes they are going to be born with; which is calls the "veil of ignorance". Assuming that everyone is rational, they need to now come up the principles of justice and fairness. He presents two principles that would be chosen by these rational people in the presented scenario - first, equality in distribution of rights and duties, and second, compensating benefits for the least advantaged members of the society.

As researchers tend to pay more attention towards fairness in the fields of Artificial Intelligence and Machine Learning, Rawlsian fairness being quite popular, has made its appearances in quite a few articles. But surprisingly, none of these articles try to use both his principles. Hence, I would like to try and see how various fainess metrics [3] perform when a classifier is trained keeping in mind both the Rawlsian principles.

II. Related work

Now I will try to look at how Rawlsian fairness has been adopted in various algorithmic decision making, especially in Artificial Intelligence and Machine Learning based algorithms.

In their recent work Shah et al. [4] introduce Rawls Classifier which is the optimal solution to an objective function that simultaneously satisfies both the properties of Pareto-efficiency (performance cannot be improved for one subgroup without harming that of another subgroup) and least-difference principles (it is okay to have some inequality given that removing this inequality will have devastating effects). If the performance metric is accuracy then the Rawls classifier minimises the error rate of the worst off subgroup over all classifiers.

Heidari et al. [5] propose a metric to evaluate decision making systems from Rawlsian "behind the veil". They introduce a benefit function and subsequently formulate a utility function for a risk-averse population. They use this utility function bounded by a constant as their constraint to the convex optimisation problem of loss minimisation. They observe that lower bounding their utility function leads to low inequality.

In one of their tutorials Gummadi et al. [6] talk about how concepts of economic inequality measurement could be extended to algorithmic fairness. Here the authors briefly talk about how interpretation of the social welfare functions as distributive justice behind a veil of ignorance motivates the idea of adopting it for algorithmic fairness measurement. In a related work [7] they talk about how algorithmic fairness measures like predictive value parity and equality of odds are just special cases of economic models of Rawlsian Equality of Opportunity.

In [8], Hashimoto et al. argue that accuracy based models can shrink the minority group over time making even initially fair models unfair over time. They propose Distributionally Robust Optimisation (DRO) which provide an upper bound on individual group risks without explicitly viewing the group identities. They optimise over this upper bound to make sure that the worst case risk is controlled over all steps. This in turn is analogous to the second Rawlsian principle.

Liu et al. [9] introduce Fair Equality of Opportunity constraints into Bayesian Networks, ie. they makes sure that the target prediction quantity in a Bayesian network is independent of the sensitive features given variables that are morally justifiable for inequality (eg. talent); which they call RAWLSNET. For this they arrives to a set of linear equations that can be solved using preexisting solvers. They use RAWLSNET's updated Bayesian Net to generate aspirational data (data that models ideally fair circumstances) that could be used to determine non-data factors in a biased system.

In another work Joseph et al. [10][11] solve the problem of Contexual Bandits - system of a predictor which sequentially predicts who to give loans to and updating itself based on who has repaid. They call this system Rawlsian fair, if an individual having higher chances of getting a loan than another person implies that they have more true quality (ie. ability to pay back) than someone. They use interval chaining to upper bound the cost of fairness. who doesn't get the loan

III. Main Idea

In the previous section we see that most real world implementations of Rawlsian fairness either uses his first principle [5][6][7][9][10][11] or second principle [4][8] but never both. This could be because of the contradicting nature of the two principles. I call it contradicting because when one introduces compensatory measures for any group, it violates equality of opportunity; and at the same time introducing only equality of opportunity would forbid compensatory discrimination.

Even though these principles may seem contradictory in the same cycle, I believe that we could apply them in subsequent steps without any problem. Consider a gradient based optimisation method which is optimising a cost function $\frac{1}{N}\sum_i l_i$ with respect to the model parameters w, where l_i is the loss for the i^{th} sample and N is the number of samples. We also have M predefined groups G_j's. Then we can define Rawlsian Gradient Descent (RGD) as a two stepped iterative process -

$$w'_t = w_t - \alpha \frac{1}{N} \sum_i \frac{\nabla l_i}{\|\nabla l_i\|} \tag{1}$$

$$\begin{aligned} w_{t+1} &= w'_t - \alpha \frac{1}{|G_k|} \sum_{i \in G_k} \nabla l_i \\ \text{where } k &= \arg \max_{j \in \{1,\dots,M\}} \frac{1}{|G_j|} \sum_{i \in G_j} l_i \end{aligned} \tag{2}$$

We repeat these steps till it either we converge according to some predefined criterion, or keep oscillating in the same region of loss space. Here eq. (1) has gradient normalisation which makes sure that the feedback from everyone in the population is uniform while eq. (2) makes sure that there is a compensatory feedback from the worst off group. We also can play around with the frequency at which eq. (1) and eq. (2) are repeated in each cycle, eg. say we could do c eq. (2) updates for every eq. (1) update.

Another way could be to formulate this as a boosting techniques like AdaBoost [12], where instead of boosting the weights for the probability of picking the worst off individual elements we could boost the weights of the worst off groups instead.

IV. Experimentation and Evaluation

It would be interesting to see how an ideal theory like Rawlsian fairness works in more practical situations.

A. Dataset

The idea here is to test the proposed method as an ablation study on UCI Adult Data Set[13]. This dataset consists of the fields 'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country', 'income'. It is a binary classification task where one need to classify a sample data point to have an 'income' in US Dollars of '$\leq$50K' or '$>$50K'. The two sensitive attributes in this dataset are 'race' and 'sex'. I used all the fields for the training. The discrete fields were converted into one hot vectors while the continuous fields were normalised to have zero mean and unit variance with respect to the training data. The cleaned training data consisted of $\sim$30K samples while the test data consisted of $\sim$15K samples.

B. Metrics

The evaluation would be done on various commonly used fairness metrics [3] like -

- **equalised odds** - the property that given a label and an attribute, the classification result is equally likely to be the chosen label for all values of the chosen attribute. Formally $P(R = +|Y = y, A = a) = P(R = +|Y = y, A = b) \quad y \in \{+, -\} \quad \forall a, b \in A$, where A is the sensitive attribute, Y is the target label, and R is the model prediction. (Note that I would be only reporting these results only for the positive targets for conciseness.)
- **predictive parity** - the property that the precision rates are equal for all the groups. Formally, $P(Y = +|R = +, A = a) = P(Y = +|R = +, A = b) \quad \forall a, b \in A$
- **demographic parity** - the property that the classification results are independent of a sensitive attribute. Formally, $P(R = +|A = a) = P(R = +|A = b) \quad \forall a, b \in A$

Normally, to improve a certain metric, people tend to optimise over the same metric. But here we would like to optimise based on the Rawlsian principles rather than any chosen metric and down the line see the behavior of the various chosen fairness metrics.

C. Experimentation

Intuitively, with small enough learning-rate eq. (1) should tend to lead to the optimal solution. And, eq. (2) would make sure that no group is left behind. From a utilitarian perspective this would lead us to a less optimal solution but the hope is that it leads to a fair solution with high amount of utility even if it is not the most optimal solution.

Following are the sets of experiments that I would like to conduct -

1) Baseline - to make sure that we are able to compare things, we should first optimise using Vanilla Gradient Descent.

2) Only eq (1) - next we should optimise using only eq (1). This will give us an idea as to how well this step works as well as what it lack.
3) Only eq (2) - this is similarly to check what the compensation step leads us to.
4) Putting it all together - here we will test out the proposed method - Rawlsian Gradient Descent - as a whole and see how it performs on out utility function as well as the fairness metrics.
5) Pretrained fine-tuning - here we can start from a model pretrained for getting good accuracy and then train the model using Rawlsian Gradient Descent there onwards.

There are a few more things that might be needed to be taken into account. First, whether the learning rate in the eq (1) and eq (2) should be the same or not. Second, for how many steps of eq (1) should there be a step of eq (2).

D. Model

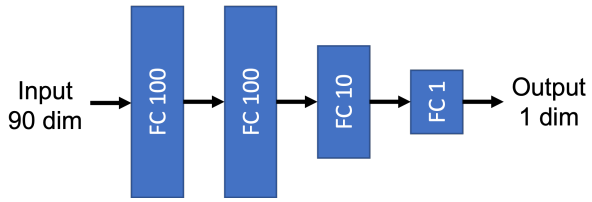

Fig. 1. Network Diagram. It is a fully connect 4 layered neural network. The input dimension is 90 and output dimension is 1. Except the last layer all the layers have ReLU activation function while the last layer has Sigmoid activation function as the problem is a binary classification.

These experiments were performed in Python using Pytorch framework. I used deep neural networks as the model for these experiments. Note that it is not necessary to use artificial neural networks as the model and this can work with any model that can be solved using gradient based optimisation. Finally, I used GPUs for these experiments which are known to accelerate the speed of optimisation in neural networks by a factor of at least 10x [14].

The network used has been described in Fig. 1. I also tried more deeper models but they didn't provide any significant improvement in the performance. I used binary crossentropy as the loss function because the task is a binary classification.

V. Results

All results reported in this section are on the test set. The results are shown were with 10 epochs and 0.1 learning rate. I tried more epochs but that did not improve the accuracy. I tried various learning rates which gave poorer results. I also tried running the two steps at different learning rates which again gave worse results than the current setting. I will be using the notation RGD(k_1:k_2) to denote that eq(1) is run k_1 times for every k_2 times eq(2) is run.

I should note that for comparing the metrics it is not enough to just get the difference of the group with the maximum metric value with the group with the least metric value. Even though, this could be useful, it might be misleading at times when the model accuracy is low but the difference in metric among groups looks quite small. Another misleading case is when the metric values themselves are quite low and hence the difference is also low. Hence, it would be better if we looked at the accuracy as well as the metric difference to make sure that a setting works well.

Another thing is that, the 'race' attribute has 5 categories in the dataset and hence tabulating the data makes it quite huge. So, instead I would be noting the maximum and minimum metric values rather than all the 5 group values.

A. Baseline vs RGD

The test accuracy is not very high even with SGD. It is usually in the range of 80-85%. All the results online on kaggle using this dataset [15] where usually using a non gradient based approach like Random Forests, etc. These methods also had their accuracies in the same range. One possible reason that I thought could be the cause was that the data was not balanced with respect to the target label. There was an imbalance of 1:3 among the binary classes. Looking at this I tried balancing the target labels and then training using SGD but that too did not improve the test accuracies. From here I concluded that maybe the dataset is hard to model. So, the rest of the experiments were conducted with the original unbalanced data.

It can be seen in Table I and Table II that RGD(1:1) performs worse than SGD for both accuracy and fairness metrics.

B. Varying k_1 and k_2 in RGD

As discussed earlier, k_1 and k_2 are hyperparameters that can be tuned. I tried running them at various ratios as can be seen in Table I and Table II.

Here RGD(1:0) and RGD(0:1) represent using only eq(1) and only eq(2) respectively. Surprisingly, one can see that only using eq(2) ie. RGD(0:1) works quite well for both 'sex' and 'race'; it leads to better Equalised odds and similar Demographic parity while having a little increase in Predictive Parity and barely harming the accuracy. On the other hand RGD(1:0) doesn't perform well on 'sex' but performs well on 'race' in terms of equalised odds.

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		Female	Male	diff	Female	Male	diff	Female	Male	diff
SGD	85.25	55.48	65.96	10.48	73.57	72.33	1.24	8.55	28.24	19.70
RGD(1:1)	83.49	52.60	66.62	14.02	67.20	67.03	0.17	8.87	30.79	21.91
RGD(0:1)	84.07	54.94	51.77	**3.17**	69.70	76.53	6.82	8.94	20.95	12.02
RGD(1:0)	84.38	39.32	57.43	18.11	80.51	74.34	6.17	5.54	23.93	18.39
RGD(1:2)	82.30	18.31	38.31	19.99	93.58	82.02	11.56	2.22	14.47	12.25
RGD(1:5)	78.74	10.41	14.41	4.00	98.31	97.42	0.89	1.20	4.58	3.38
RGD(1:10)	83.92	31.78	57.27	25.49	83.10	73.11	9.99	4.34	24.26	19.93
RGD(2:1)	79.77	78.64	82.50	**3.87**	46.70	58.01	11.31	19.09	44.05	24.96
RGD(5:1)	83.86	63.38	65.54	**2.17**	63.72	68.62	4.90	11.28	29.59	18.31

TABLE I
RESULTS FOR 'SEX' ATTRIBUTE : BASELINE AND RGD (ALL NUMBERS ARE IN PERCENTAGE)

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		max	min	diff	max	min	diff	max	min	diff
SGD	85.28	66.12	36.84	29.27	85.71	69.75	15.97	25.74	6.04	19.70
RGD(1:1)	77.37	8.46	0.00	8.46	100.00	0.00	100.00	2.20	0.00	2.20
RGD(0:1)	85.31	73.68	50.00	**23.68**	85.71	59.38	26.34	30.88	11.48	19.41
RGD(1:0)	84.36	63.64	44.64	**18.99**	86.67	67.54	19.12	27.94	7.38	20.56
RGD(1:2)	81.94	33.17	0.00	33.17	100.00	0.00	100.00	9.74	0.00	9.74
RGD(1:5)	81.85	78.89	2.98	75.91	100.00	60.88	39.12	33.65	0.35	33.29
RGD(1:10)	83.35	73.01	2.38	70.63	100.00	65.42	34.58	28.98	0.28	28.70
RGD(2:1)	79.65	85.36	4.17	81.20	100.00	55.94	44.06	39.62	0.67	38.95
RGD(5:1)	84.08	57.10	4.17	52.93	100.00	74.28	25.72	19.96	0.67	19.29

TABLE II
RESULTS FOR 'RACE' ATTRIBUTE: BASELINE AND RGD (ALL NUMBERS ARE IN PERCENTAGE)

In general, in most other choices of k_1 and k_2, RGD seems to not perform any better than SGD. In fact, it seems like adding the eq(1) seems to make it worse for sensitive attribute 'race' based training. But RGD(2:1) and RGD(5:1) lead to quite good Equalised odds for the case of 'sex'.

C. Pretraining + RGD

Another experiment that I did was to pretrain the model using SGD for 10 epochs before running RGD on it. The intuition was to start from a good point in the optimisation landscape to start further fairness based optimisation. But this did not give better results than SGD. These results can be found in the Appendix A's Table V and Table VI.

D. Removing normalisation from Eq (1)

The reason that I had introduced the normalisation in the eq (1) had no logical basis but was entirely based on the Rawlsian principle 1 that everyone's opinion should have the same weight. Hence here normalisation meant that the gradients due to each sample would have unit norm. But since the noramalisation step doesn't work very well, hence I thought of removing the normalisation from the step. This means that it will get converted to eq (3), which is nothing but the same as SGD.

$$w_t' = w_t - \alpha \frac{1}{N} \sum_i \nabla l_i \tag{3}$$

Now, we will instead run eq (3) and eq (2) alternatively instead. The results from this experiments have been reported in Table III and Table IV. It can be seen that this experiments worked quite well. Even though in the best performing settings, the metric differences have not gone down a lot with respect to the baseline but the metric's value for the maximum group and the minimum group has improved significantly. In other words the metric for all the groups have improved in this setting. One can observe that RGD(1:5) works quite well for both the settings. Here the Equalised odds are better for both 'race' and 'sex' while the Predictive Parity and Democratic Parity don't have much change compared to baseline.

E. Pretraining + Removing normalisation from Eq((1)

Since the previous experiment worked better than baseline, hence I thought that maybe if we could give it a pretrained model then it might have a better performance. The results have been reported in the Appendix A in Table VII and Table VIII.

In the case of 'sex', Pretrained + Removing normalisation doesn't work much better than SGD. For 'race' the results close to SGD. This is maybe because pretraining makes the state to stay close to the SGD final state.

VI. LIMITATIONS

This study assumes that there are predefined disjoint sets of groups while implementing eq. (2). This is

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		Female	Male	diff	Female	Male	diff	Female	Male	diff
RGD(1:1)	84.89	58.17	69.93	11.76	70.43	69.56	0.88	9.36	31.14	21.78
RGD(1:2)	85.37	49.55	61.37	11.82	78.63	75.26	3.37	7.14	25.26	18.11
RGD(1:5)	84.62	61.04	70.38	**9.34**	67.59	68.76	1.17	10.24	31.70	21.47
RGD(1:10)	83.98	53.68	69.42	15.74	65.86	67.76	1.91	9.24	31.73	22.49
RGD(2:1)	84.64	62.48	70.44	**7.96**	67.44	68.74	1.29	10.50	31.74	21.24
RGD(5:1)	84.20	50.99	60.55	9.56	73.20	71.41	1.79	7.90	26.26	18.37

TABLE III
RESULTS FOR 'SEX' ATTRIBUTE WITH NORAMLISATION REMOVED FROM RGD EQ (1) (ALL NUMBERS ARE IN PERCENTAGE)

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		max	min	diff	max	min	diff	max	min	diff
RGD(1:1)	85.36	68.60	41.67	26.93	83.33	70.99	12.34	26.72	9.28	17.43
RGD(1:2)	84.63	56.20	15.79	40.41	90.91	72.12	18.79	21.57	2.68	18.88
RGD(1:5)	81.86	80.64	58.33	**22.31**	73.68	57.35	16.33	35.16	12.75	22.41
RGD(1:10)	83.43	62.81	42.11	**20.70**	92.31	64.84	27.46	27.70	6.71	20.98
RGD(2:1)	84.00	76.86	52.63	**24.23**	87.50	62.09	25.41	32.11	8.72	23.38
RGD(5:1)	84.16	77.69	47.37	30.32	86.67	62.79	23.88	35.29	8.72	26.57

TABLE IV
RESULTS FOR 'RACE' ATTRIBUTE WITH NORAMLISATION REMOVED FROM RGD EQ (1) (ALL NUMBERS ARE IN PERCENTAGE)

generally not true in the real world, where by default a lot of intersectionality seeps into picture. But that is out of scope of this study as it tries to replicate the fairness theory by Rawls which itself has the same set of underlying assumption. But never the less, it could be included in future work.

VII. DISCUSSION

One question that might arise is why do we not just optimise on the metrics that we are measuring. It is quite evident in a lot of cases that various fairness metrics may not monotonically increase with other metrics. In fact in a lot of cases there is a trade off among these. Hence the idea here is to find a solution that tries to do well on all these metrics without explicitly using them into the optimisation process. If we try to fix the metrics and optimise on them then it might do well on just that metric while performing poorly on others. At the same time if we introduce too many metrics to optimise on, the optimisation may not converge. Further, there might be a scenario that a chosen metric is suitable for a given application and not for another one and in such cases you would need to change the pipeline for every application. But doing the optimisation in the proposed way solves this problem.

It was observed in the previous section that just using RGD didn't work very well but only using Eq (2)[Compensation] worked quite well. An intuition behind this could be that RGD(0:1) is quite similar to online hard mining ie. only sending the gradient updates for the hardest examples which is quite widely used in the literature[16]. But here instead we are not just choosing the hardest samples from all the groups but instead choosing the hardest group itself to send the gradients back for.

We saw that RGD with Eq (3)[Equal Opportunity] and Eq (2)[Compensation] worked quite poorly. The probable reason is that the normalisation in the Eq (3)[Equal Opportunity] does not lead to any benefit from the optimisation point of view and rather harms the optimisation process when used along with Eq (2)[Compensation]. Next, one might ask why did RGD(1:5) with Eq (3)[SGD] and Eq (2)[Compensation] work well. As discussed earlier, Eq (3)[SGD] on itself works quite well with respect for fairness metrics but having Eq (3)[SGD] improves the optimisation and hence the accuracy and it in turn increases the fairness metrics for all the groups. In total the new methods is like SGD with Hard Group Mining.

VIII. CONCLUSION AND FUTURE WORK

There are a lot of ML models that are proving to be biased which has led to various fairness measures being introduced at the various stages in the ML pipeline. There are various theories of fairness and justice that exist and this was a test to see if one of these theories can be introduced into the optimisation stage of the ML pipeline. It was observed that although RGD as initially defined did not work better than the baseline, with a minor modification it starts working better than the baseline for multiple sensitive attributes on the chosen dataset.

This work can also be extended to other datasets like German Credit Data Set [17]. Unfortunately, I did not get enough time to analyse that dataset.

REFERENCES

[1] L. Wenar, "John rawls‖ stanford encyclopaedia of philosophy," *First published Tue Mar 25, 2008, retrieved 18 th April 2010*, 2008.

[2] J. Rawls, *A theory of justice: Revised edition*. Harvard university press, 1999.

[3] *Machine Learning Glossary: Fairness — Google Developers*, en. [Online]. Available: https : / / developers . google . com / machine - learning / glossary / fairness # fairness - metric (visited on 10/25/2021).

[4] K. Shah, P. Gupta, A. Deshpande, and C. Bhattacharyya, "Rawlsian fair adaptation of deep learning classifiers," *arXiv preprint arXiv:2105.14890*, 2021.

[5] H. Heidari, C. Ferrari, K. P. Gummadi, and A. Krause, "Fairness behind a veil of ignorance: A welfare analysis for automated decision making," *arXiv preprint arXiv:1806.04959*, 2018.

[6] K. P. Gummadi and H. Heidari, "Economic theories of distributive justice for fair machine learning," 2019, pp. 1301–1302.

[7] H. Heidari, M. Loi, K. P. Gummadi, and A. Krause, "A moral framework for understanding fair ml through economic models of equality of opportunity," in *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 181–190.

[8] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness without demographics in repeated loss minimization," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1929–1938.

[9] D. Liu, Z. Shafi, W. Fleisher, T. Eliassi-Rad, and S. Alfeld, "Rawlsnet: Altering bayesian networks to encode rawlsian fair equality of opportunity," *arXiv preprint arXiv:2104.03909*, 2021.

[10] M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Fair algorithms for infinite and contextual bandits," *arXiv preprint arXiv:1610.09559*, 2016.

[11] M. Joseph, M. Kearns, J. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," *arXiv preprint arXiv:1605.07139*, 2016.

[12] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[13] H. Hofmann, *UCI Machine Learning Repository: Adult Data Set*. [Online]. Available: https : / / archive.ics.uci.edu/ml/datasets/adult (visited on 10/27/2021).

[14] *CPU x10 faster than GPU: Recommendations for GPU implementation speed up*, en, Sep. 2019. [Online]. Available: https://discuss.pytorch.org/t/cpu-x10-faster-than-gpu-recommendations-for-gpu-implementation-speed-up/54980 (visited on 11/14/2021).

[15] Anirudhraj, *Adult salary predictor*, May 2020. [Online]. Available: https://www.kaggle.com/anirudhraj/adult-salary-predictor.

[16] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[17] H. Hofmann, *UCI Machine Learning Repository: Statlog (German Credit Data) Data Set*. [Online]. Available: https : / / archive . ics . uci . edu / ml / datasets/statlog+(german+credit+data) (visited on 10/27/2021).

APPENDIX A
SUPPLEMENTARY MATERIAL

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		Female	Male	diff	Female	Male	diff	Female	Male	diff
RGD(1:1)	84.61	43.81	49.98	6.18	78.21	81.15	2.94	6.35	19.08	12.73
RGD(1:2)	80.15	15.62	22.18	6.56	85.29	92.32	7.02	2.08	7.44	5.36
RGD(1:5)	81.22	52.06	25.23	26.83	68.56	91.04	22.49	8.61	8.58	0.03
RGD(1:10)	77.54	9.34	8.50	0.84	98.11	99.63	1.51	1.08	2.64	1.56
RGD(2:1)	77.07	32.68	19.15	13.52	49.46	63.10	13.65	7.49	9.40	1.91
RGD(5:1)	85.17	54.40	61.37	6.98	74.63	74.45	0.18	8.26	25.53	17.27

TABLE V
RESULTS FOR 'SEX' ATTRIBUTE WITH SGD PRETRAINING (ALL NUMBERS ARE IN PERCENTAGE)

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		max	min	diff	max	min	diff	max	min	diff
RGD(1:1)	81.56	28.65	8.33	20.32	100.00	91.21	8.79	8.16	1.34	6.82
RGD(1:2)	24.56	100.00	99.94	0.06	29.66	11.91	17.75	100.00	99.98	0.02
RGD(1:5)	77.45	10.53	8.22	2.30	100.00	91.67	8.33	2.94	0.99	1.95
RGD(1:10)	84.94	61.98	25.00	36.98	78.95	66.67	12.28	23.28	4.70	18.59
RGD(2:1)	84.16	74.47	21.05	53.41	88.89	66.14	22.75	29.24	3.36	25.88
RGD(5:1)	75.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

TABLE VI
RESULTS FOR 'RACE' ATTRIBUTE WITH SGD PRETRAINING (ALL NUMBERS ARE IN PERCENTAGE)

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		Female	Male	diff	Female	Male	diff	Female	Male	diff
RGD(1:1)	85.23	55.66	62.27	6.61	75.43	73.90	1.52	8.37	26.10	17.73
RGD(1:2)	85.16	53.32	62.74	9.42	75.57	73.58	1.99	8.00	26.41	18.41
RGD(1:5)	85.05	52.78	61.98	9.20	75.58	73.59	1.99	7.92	26.09	18.17
RGD(1:10)	85.12	51.89	59.05	7.17	75.46	75.82	0.36	7.80	24.13	16.33
RGD(2:1)	85.03	58.17	67.13	8.96	70.90	71.14	0.24	9.30	29.23	19.93
RGD(5:1)	85.11	50.45	59.91	9.46	77.20	75.14	2.06	7.41	24.70	17.29

TABLE VII
RESULTS FOR 'SEX' ATTRIBUTE WITH WITH SGD PRETRAINING AND NORAMLISATION REMOVED FROM RGD EQ (1) (ALL NUMBERS ARE IN PERCENTAGE)

Method	Accuracy	Equalised odds			Predictive Parity			Demographic Parity		
		max	min	diff	max	min	diff	max	min	diff
RGD(1:1)	84.95	71.90	47.37	24.53	92.86	56.25	36.61	29.90	8.15	21.75
RGD(1:2)	85.29	71.07	26.32	44.76	90.91	66.92	23.99	28.68	4.70	23.98
RGD(1:5)	85.00	75.21	36.84	38.36	80.00	61.14	18.86	32.84	7.38	25.46
RGD(1:10)	85.04	66.94	26.32	40.63	100.00	67.41	32.59	26.72	3.36	23.36
RGD(2:1)	84.87	72.73	26.32	46.41	87.50	62.50	25.00	28.92	5.37	23.55
RGD(5:1)	85.20	60.33	15.79	44.54	90.00	70.00	20.00	22.55	2.68	19.86

TABLE VIII
RESULTS FOR 'RACE' ATTRIBUTE WITH WITH SGD PRETRAINING AND NORAMLISATION REMOVED FROM RGD EQ (1) (ALL NUMBERS ARE IN PERCENTAGE)