Fairness using Discrepancy Minimization

First Author[1*], **Second Author**[2], **Third Author**[2,3] and **Fourth Author**[4]
[1]First Affiliation
[2]Second Affiliation
[3]Third Affiliation
[4]Fourth Affiliation
{first, second}@example.com, third@other.example.com, fourth@example.com

Abstract

With the rise of the era of artificial intelligence and machine learning in the last decade, there has been an increasing interest in developing a strong theory and implementation of *Algorithmic Fairness* which has eventually resulted in a large volume of work over the past few years. Despite the huge amount of work done on the topics over a very short period, there has been little consensus of a unifying theory of algorithmic fairness. In this paper we develop a notion of fairness that is based on the notion of discrepancy of set systems, a widely studied topic in the theory of computer science and combinatorics [8], and use it to solve a weighted version of the problem of mitigating disparate impact. To the best of our knowledge this is the first use of combinatorial discrepancy in the context of fairness in machine learning.

1 Introduction

In the past decade, there has been a growing interest to develop algorithms that will make policy-oriented decisions like recruitment, selection in Universities, promotion for a job title, crime recidivism etc which has in turn led to a surge in research to define the notion of *fairness* for algorithms. This is because several algorithms for making decisions have been found to be unfair. For example, the software COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) has been found to be biased [22]. In another case of a hiring application, it was recently exposed that Amazon discovered that their AI hiring system was discriminating against female candidates, particularly for software development and technical positions [22].

Consequently, several notions of fairness in machine learning algorithms have been proposed which include disparate impact, equal opportunity, disparate treatment, equalized odds etc. [22]. Out of which disparate impact is one of the early definitions in the field of fairness. NG: First write in legal terms what is disparate impact, how is that mathematically transformed and what are the limitations in the definition MP: we have removed the term of disparate impact and only talked about demographic parity In this paper we revisit the problem of mitigating demographic parity (the difference version of disparate impact) in case of multiple and polyvalent sensitive attributes. The demographic parity for multiple sensitive attribute can be defined in several ways like considering the difference of maximum and minimum acceptance rates among all the values of sensitive attributes. One can also define disparate impact as the average of the difference of the one versus all for all the values of the sensitive attribute.

The algorithms which have been proposed to remove unfairness from the data can be broadly classified in three groups: *pre-processing, in-processing* and *post-processing* based algorithms. **Pre-processing**: The goal is to pre-process the training data such that any classification algorithm trained on this data would generate unfairness-free outcomes. This is usually done by generating fair representations as is done by Feldman et al. [14]; Dwork et al. [12]; Kamiran and Calders, [15], Edwards and Storkey [13]; Madras et al. [21]; Beutel et al. [5]. **In-processing**: Here the idea is to add constraints for fairness to the classification optimization model like SVM; examples - Calders and Verwer [6]; Kamishima et al. [16]; Bechavod and Ligett, [3] Bilal Zafar et al. [30]; Wu et al. [29], Padala and Gujar [24] and Zhang et al. [31]. **Post-processing** : The third and final strategy consists of first running a standard classifier like SVM or Logistic-regression on the training data and then use the model to mitigate unfairness in the test data. This approach has been used by Corbett-Davis et al. [9]; Agarwal et al. [1] and Narasimhan [23]. All these approaches has certain pitfalls which have been circumvented by us. One of the major pitfalls is that most of the algorithms fail to deal with multiple sensitive attributes in a memory efficient way. More formally, if the number of sensitive attributes, k increases the algorithms which can deal with multiple attributes end up performing an exponential 1-hot encoding that leads to a blow-up of datasize. This problem has been mentioned to be critical in papers like [30]. We circumvent this approach as our approach only incurs a space that is only linear in k. We would discuss these issues in more detail in Section 3.1. NG: What are the limitation of the approaches; bring it from pitfalls MP: Limitations and pitfalls are written above this and contributions are written below this

In this paper we have proposed the first post-processing fair classification algorithm that allows us to control acceptance rates across groups and we have achieved fairness vs accuracy

*Contact Author

trade-offs which are either comparable or better than prior works. We also solve the multiple (race and gender) and polyvalent (race) attribute case and the problem of **fairness gerrymandering** in the most memory efficient way that avoids exponential one hot encoding. We reformulate the problem and provide Linear Programming based solutions to the problem based on the notion of *combinatorial discrepancy* which outperform the existing algorithm under certain metrics.

In this paper we study a notion of fairness that is based on the notion of discrepancy of set systems, a widely studied topic in the theory of computer science and combinatorics [8]. To the best of our knowledge this is the first use of combinatorial discrepancy in the context of fairness in machine learning. Our paper is the first to address the issues of controlling the acceptance rates for multiple sensitive attributes. This feature has not been explored in the previous algorithm. As a consequence, we show that is feature allows us to maintain high accuracy as well the desired value of selection rates across various groups. Our framework allows us to control the acceptance rate of a particular group as well, without performing exponential one-hot encoding of sensitive attributes.

NG: Section overview

NG: Remove background and our contribution

2 Problem Definition

NG: Bring in metrics NG: A little about disparate impact and demographic parity the relation they have and then just state that you are working on demographic parity in this paper - various metric how you would measure demographic parity. MP: We have added the definitions and various metrics below **Disparate Impact**: This measure was designed to mathematically represent the legal notion of disparate impact. It requires a high ratio between the positive prediction rates of protected and non-protected groups. The constraint is formulated as

$$\mathbb{P}[\hat{Y} = a | S \neq 1] \geq \mathbb{P}[\hat{Y} = a | S = 1](1 - \Delta)$$

Here Δ is a fraction close to zero.

Demographic Parity: This measure is similar to disparate impact, but the difference is taken instead of the ratio. This measure is also commonly referred to as statistical parity.

$$\left|\mathbb{P}[\hat{Y} = a | S \neq 1] - \mathbb{P}[\hat{Y} = a | S = 1]\right| \leq \Delta$$

In this paper, we will focus on the difference version of the definition of disparate impact. More specifically, we would use the definition of demographic parity. Since in this paper we focus on the multi-attribute case, we also need to consider the possible definitions of demographic parity (henceforth referred to as DP) in this case. We can consider the first one among the following two definitions:

$$DP_1 = \max_{S,s} \mathbb{P}[\hat{Y} = 1 | S = s] - \min_{S,s} \mathbb{P}[\hat{Y} = 1 | S = s]$$

$$DP_2 = \frac{1}{|S|} \sum_{S,s} |\mathbb{P}[\hat{Y} = 1 | S = s] - \mathbb{P}[\hat{Y} = 1 | S \neq s]|$$

We formulate the problem of mitigating DP in the two different versions:

Problem 1: Given a dataset X with $|X| = n$ records, $S = \{S_1, S_2 \dots S_m\}$ be the set of sensitive attributes with S_j taking k_j possible values and parameters ϵ, β_j^i for $j \in [m], i \in [k_j]$ find a ± 1 labelling of the records such that the positive class percentage for the i^{th} value of the j^{th} sensitive attribute is between β_j^i and $\beta_j^i + \epsilon$. In a special case when all $\beta_j^i = \beta$, problem becomes reducing the disparate impact to ϵ.

Problem 2: We consider the same set up as above with same input parameters and in addition we are given y_i for $i \in [n]$ the actual labels of the records in the dataset, along with a parameter a and we have to find a labelling that satisfies all the constraints of Problem 1 and the accuracy of the labelling w.r.t the actual labels is more than a.

2.1 Pitfalls of Existing Approaches

NG: make it much shorter MP: below is a shortened version Firstly, the definition of disparate impact (that is defined for a single sensitive attribute with two values like Male, Female) doesn't naturally extend for multiple values. Secondly, if $S = \{S_1, S_2 \dots S_m\}$ be the set of sensitive attributes with S_j taking k_j possible values, then the existing algorithms perform a 1-hot encoding of these to work with a sensitive datasize and number of fairness constraints of

$$n\left(\prod_{j=1}^{m} k_j\right) \quad \text{and} \quad 2\left(\prod_{j=1}^{m} k_j\right)$$

which is an exponential increase in terms of the size of the number of sensitive attributes. It has been noticed by [30] that such a blow-up is needed to avoid the so called **Fairness Gerrymandering** [17] problem which arises when one fails to achieve uniform positive class selection rate across all possible subgroups (for eg. "black male from Africa with age ¿ 35"). It is not clear that solving the fairness gerrymandering problem is indeed useful in realistic scenarios. We propose that addressing it to some extent might be useful if we do not incur a exponential blow up. A possible way we can use out framework to deal with the gerrymandering problem is to ask the user to enumerate some groups $G_1, G_2, \dots G_k$ such that each $G_i \in X$ represents a group consisting of several sensitive attributes (like "Male Black American with Age ¿ 45") and ask for equal acceptance rates (β) for these groups. Such an input can be easily handled with in our system as our problem formulation takes sets as input which is not the case with existing notions of fairness. Our framework works with a sensitive datasize and number of fairness constraints

$$n\left(\sum_{j=1}^{m} k_j\right) \quad \text{and} \quad 3\left(\sum_{j=1}^{m} k_j\right)$$

respectively.

3 Discrepancy as a Measure of Fairness

NG: Demographic Parity evaluated through the lens of discrepancy theory In this section we discuss about discrepancy theory that is a widely study topic in the field of mathematics and theoretical computer science. We then evaluate demographic parity through the lens of combinatorial discrepancy theory.

3.1 Discrepancy Theory

Discrepancy theory is an important topic of study that basically originated from questions in analytic number theory which were asked by pioneer mathematicians like van der Corput [10], Schmidt and Erdos [27] in early Twentieth century who studied the combinatorial and analytic properties of infinite sequences of integers in a particular range. This lead to the development of several variants and questions of number theory which were collectively called discrepancy theory. In computer science, nowadays there has been extensive use of discrepancy theory including areas in complexity theory, probabilistic algorithms, pseudo-randomness, computational geometry, machine learning, communication complexity, mathematical finance and computer graphics [8]. In this paper we will develop an application of discrepancy theory in the newly developing field of algorithmic fairness.

3.2 Combinatorial Discrepancy

In this paper, we mostly deal with the notion of combinatorial discrepancy [8] that is defined for set-systems in combinatorics. Given a universal set X consisting of n elements and a collection $S = \{S_1, S_2 \dots S_m\}$ of subsets of X. The objective is to find a coloring $\chi : X \to \{-1, 1\}$ such that the *discrepancy* of the set system is minimized. Where the discrepancy of a particular set S_j w.r.t. a coloring χ is defined as

$$D(S_j) = \Big| \sum_{a \in S_j} \chi(a) \Big|$$

Our aim is to find

$$\begin{aligned} D(X, S) &= \min_{\chi \in \{-1,1\}^n} \max_{S_i \in S} D(S_i) \\ &= \min_{\chi \in \{-1,1\}^n} \max_{S_i \in S} \Big| \sum_{a \in S_j} \chi(a) \Big| \end{aligned}$$

This problem has been studied from both algorithmic and hardness point of views for over two decades and very recently there has been a plethora of results obtained for this problem because of emphasis on its research in theoretical computer science community. More formally, in a very seminal paper Spencer [26] showed that there always exist a coloring that achieves a discrepancy of within $6\sqrt{n}$. This result is called the six-standard deviation result. Although this result is non-constructive, constructive versions started coming up beginning from the work of Bansal [2] who gave an SDP based algorithm to find a coloring of low discrepancy. This result was simplified by Lovett and Meka [20]; several constructive results appeared after these work most importantly the work of Larsen [19] who provided an implementable (in Python) algorithm to find a low discrepancy coloring. Although theoretically his result isn't as sound as previous constructive algorithms, the implementability of his algorithm for large values of n makes it more important in the machine learning set-up where we are interested in implementation of algorithms and not just theoretical guarantees. Larsen's algorithm finishes in a reasonable amount of time on matrices of sizes up to 10000×10000 where the matrix is the matrix corresponding to the adjacency properties of the given set system. Some hardness results are also known for this problem [7].

3.3 Discrepancy as Fairness

In this section we describe how the notion of combinatorial discrepancy can be used to define a notion of fairness for machine learning algorithms. Given a particular dataset we first create an instance of the discrepancy problem by defining (X, S) as follows: X will be the set of records in the dataset s.t. $|X| = n$ and for each sensitive attribute i (like race, sex , religion) we construct $S_i^j \in S$ which consist of all the elements of X that have a fixed value of j of the sensitive attribute i. For example if i is 'sex' then there can be two possible values of j corresponding to 'Male' and 'Female'; S_i^1 will consists of all the elements of X which are Male and S_i^2 will consists of the females. Thus we have created an instance of the discrepancy problem. We can also represent the collection S of sets by a $k \times n$ matrix R whose rows represent the characteristic vector of each $S_i \in S$.

Meaning of a Coloring: As we have seen that there is a natural instance of the discrepancy problem for a given dataset with certain sensitive attributes. Now we need to understand the meaning of a low discrepancy coloring for the set system obtained as above and how we can interpret it as a notion of fairness. Intuitively, a low discrepancy coloring will balance the colors for each of the categories of the sensitive attributes. For example if a coloring represents whether a person gets a job (+1) or not (-1) then a low discrepancy coloring will ensure that the ratio of selected candidates to non-selected candidates for each of sensitive attributes will be close to $0.5 : 0.5$. More formally, if S_j^i represents the set of all candidates whose value of j^{th} sensitive attribute is i a low discrepancy coloring will ensure

$$\left| X_p^{ij} - X_n^{ij} \right| \leq \epsilon$$

where $X_p^{ij} = |\sum_{a \in S_j^i, \chi(a)=1} \chi(a)|$ and $X_n^{ij} = |\sum_{a \in S_j^i, \chi(a)=-1} \chi(a)|$. As we had noted that this definitions enforces a 50/50 split for all the possible choices of the sensitive attributes. In order to enforce a $\beta : 1 - \beta$ split one can consider a *weighted version* of the discrepancy problem as discussed in [11]. In the weighted version of the problem we try to ensure

$$(\delta)|S_j^i| \leq \left| X_p^{ij} - X_n^{ij} \right| \leq (\delta + \epsilon)|S_j^i|$$

where $\delta = 2\beta - 1$ and $\epsilon > 0$ is a parameter close to 0. We can also work with an equivalent notion of bounding $\frac{X_p^{ij}}{X_n^{ij}}$. We will consider in detail of algorithms to generate colorings with such guarantees in the next section in the case of polyvalent and multiple sensitive attributes.

3.4 Equivalence of Discrepancy and Disparate Impact

In the following we prove that in the case when the data has one sensitive attribute which can take two values; the notions of discrepancy and disparate impact are equivalent upto constant factors. Let X be a dataset with a sensitive attribute S which can take two possible values $0, 1$ and the number of

records in X with $S = 1$ is η and with $S = 0$ is γ (w.l.o.g we assume $\gamma \geq \eta$). Let $\chi : X \to \{1, -1\}$ be a binary labelling (classification) of X which has discrepancy of $\mathbf{D}_\chi$ and demographic parity of $\mathbf{DP}_\chi$ Then we can prove the following result.

Lemma 1

$$\frac{1}{2}\left(\frac{1}{\eta} - \frac{1}{\gamma}\right)\mathbf{D}_\chi \leq \mathbf{DP}_\chi \leq \frac{1}{2}\left(\frac{1}{\eta} + \frac{1}{\gamma}\right)\mathbf{D}_\chi$$

Lemma 2 *As $n \to \infty$ (the number of records in the data is large), $\mathbf{DP}_\chi = k\mathbf{D}_\chi$ for some constant k independent of n.*

The results can be proven using applying the definitions of demographic parity and combinatorial discrepancy and simple algebraic manipulation.

4 Mitigating Demographic Parity Using Discrepancy

In this section, we propose several techniques to mitigate disparate impact from the data by using the notion of combinatorial discrepancy. Our approach is similar to that of [14] in which they introduce a notion of *balanced error rate* (BER) to reduce disparate impact in the data.

4.1 Linear Programming Based Approach

In this section we describe how can we use a linear programming relaxation for the problem of discrepancy minimization. Here we use the solution of an LP to arrive at a coloring of low discrepancy. Although in general the discrepancy problem is NP-hard, we are interested in getting a LP relaxation to the problem that can give us a coloring of low enough discrepancy if not exact. This optimization problem is an integer linear programming that involves the minimization of a modulus function and it is known that such an optimization can be relaxed to linear programming problem as follows: we introduce a new variable to the system say z and perform the following optimization.

LP-1

$$\min z$$
$$z \geq \sum_{a \in S_j} \chi(a) \ \forall S_j \in S$$
$$z \geq -\sum_{a \in S_j} \chi(a) \ \forall S_j \in S$$
$$z \geq 0$$
$$-1 \leq \chi(a) \leq 1 \ \forall a \in X$$

We will call this linear program **LP-1**. This LP relaxation has been studied extensively from the theoretical point of view in the so called Beck-Fiala [4] setting for discrepancy with the aim of getting relevant theoretical bounds for the problem. But here we are more interested in its implementability for large number of variable. We use well known libraries of Python like PuLP to solve the above Linear Programming problem.

NG: The goodness of LP is that it is not restricted to 50% but you can extend it to any arbitrary acceptance MP: We have written about the need of weighted discrepancy in the following section

Weighted Discrepancy

As we had discussed in the previous section, we can also conceive of applications in which we need to control the discrepancy rather than minimizing it. Think of an examination in which we want to enforce that the ratio of selected and non-selected candidates is not 1 but a very small number like 0.01. This scenario arises in several highly competitive examinations (SAT, JEE etc.) [18]. In these scenarios it won't be a good idea to find a coloring that minimizes discrepancy because that may lead to a a 50/50 split (or a ratio close to 1). Thus in order to retrieve a coloring that can solve the weighted discrepancy problem with a $\beta_j : 1 - \beta_j$ split for the j^{th} sensitive attribute (in a special case all β_j's can be made equal to β which implies 0 disparate impact), we can use the following variant of the above linear program.

LP-2

$$\min 1$$
$$\sum_{a \in S_j} \chi(a) \geq (2\beta_j - 1)|S_j| \ \forall S_j \in S$$
$$\sum_{a \in S_j} \chi(a) \leq ((2\beta_j - 1) + \epsilon)|S_j| \ \forall S_j \in S$$
$$-1 \leq \chi(a) \leq 1 \ \forall a \in X$$

We will call this optimization problem **LP-2**. This LP allows us to vary $\beta_j > 0$ and ϵ to control the acceptance rates across groups and the error rate respectively.

NG: We can talk about constraint in the experimental section MP: feasibility discussions have been pushed to experimental section

5 Demographic Parity With Accuracy

5.1 Randomized Approach

Definition 1 *Let A_1 be a binary classification algorithm on a dataset D and A_2 be an algorithm that returns a coloring of discrepancy bounded by Δ of the corresponding instance of D. We generate new labels called Fair$(A_1, A_2, \alpha, \Delta)$ labelling, if on all the records in which A_1 disagrees with A_2, with probability α we choose the label assigned by A_2 and with probability $1 - \alpha$ we choose the labelling of A_1.*

As with the prior work on algorithmic fairness it has been noticed that a trade-off w.r.t *accuracy* arises when we try to implement a notion of fairness. We try to develop similar trade-offs in the above definition as well. Since α measures the amount of fairness in the final labelling, we need to derive a relation between α and Acc_f which is the accuracy of the final labelling. The following result justifies that.

NG: so is this lemma even required This Lemma justifies the accuracies obtained after the random mixing of colorings of svm and our LP

Lemma 3 $\mathbb{E}[Acc_f] = (1-\alpha)Acc_{A_2} + \alpha Acc_{A_1}$ *where* Acc_{A_2} *is the training accuracy of* A_2 *and* Acc_{A_1} *is the accuracy of* A_1. *Thus the expected accuracy of the final coloring is a linear combination of the accuracies of* A_1 *and* A_2.

Since Fair$(A_1, A_2, \alpha, \epsilon)$ is a randomized algorithm, that returns a final binary (± 1) labelling f on the records that has actual labelings as r, Acc$_f$ is a random variable and we need to find its expected value that is equal to

$$\begin{aligned}
\mathbb{E}[\text{Acc}_f] &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\mathbb{1}f(x_i) = r(x_i)] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}[f(x_i) = r(x_i)] \\
&= \frac{1}{n}\sum_{i=1}^{n}\{\alpha\mathbb{1}[A_1(x_i) = r(x_i)] + \\
&\quad (1-\alpha)\mathbb{1}[A_2(x_i) = r(x_i)]\} \\
&= \alpha\{\sum_{i=1}^{n}\mathbb{1}[A_1(x_i) = r(x_i)]\} + \\
&\quad (1-\alpha)\{\sum_{i=1}^{n}\mathbb{1}[A_2(x_i) = r(x_i)]\} \\
&= (1-\alpha)\text{Acc}_{A_2} + \alpha\text{Acc}_{A_1}
\end{aligned}$$

NG: I dont see any section where you have talked about discrepancy set and how that helps.So a section about discrepancy set is required

Done in starting of section 5 NG: Such hard coding is not acceptable

LP with Accuracy Constraint

NG: LP-3 is not clear, also if you put accuracy constraint here, then what is the need for the randomization, also the accuracy part is not clearly explained. answer: Even after adding accuracy constraints the accuracy of the LP is not that much(example : for adult data it's is about 81 percent with 0.05 di for) to boost that accuracy to about 82 % or more we need randomization.

As described in the aforementioned sections, the linear programs **LP-1** and **LP-2** are not designed to improve the accuracy of the classifier as well. To that end, we design a new LP which we call **LP-3** in which we use the predictions r_i of a Linear SVM (as pseudo real labels). More formally, we perform the following optimization problem in the "test data"".

LP-3

$$\min 1$$

$$\sum_{a\in S_j}\chi(a) \geq (2\beta_j - 1)|S_j| \;\forall S_j \in S$$

$$\sum_{a\in S_j}\chi(a) \leq ((2\beta_j - 1) + \epsilon)|S_j| \;\forall S_j \in S$$

$$\sum_{a\in D_t}\chi(a)r(a) \geq \delta|D_t|$$

$$-1 \leq \chi(a) \leq 1 \;\forall a \in X$$

NG: I dont see any result about LP-3, what are the two approaches?? This is not clear For results refer to figure:8 and we will add more related / relevant text in result section.

NG: This section is not needed

6 Datasets

6.1 Real Datasets

In our study we have used 4 different data set such as ProPublica risk assessment, ProPublica violent risk assessment [25] and Adult Income dataset [28]. All results of SVM implementation are obtained using 70%/30% train/test splits.

ProPublica recidivism The ProPublica data is obtained from COMPAS risk assessment system. It includes 6167 records along with 13 attributes for and The target variable for each criminal to predict whether criminal recidivated (rearrested) or not within two years from his/her last release from the prison. The data have attributes such as crime charge degree, decile score, score text etc. with sensitive attributes gender and race. NG: What about if we consider the other attributes as also sensitive attribute

ProPublica violent recidivism The violent recidivism is another version of ProPublica data. The data having records of 4010 criminals with same attributes as of ProPublica data. The target variable is whether the criminal is rearrested or not within two years only for violent crimes. The sensitive attributes are race and gender same as for ProPublica data.

Adult The UCI Adult dataset contains information of 1994 census data is one of the largest used dataset for fairness related studies. The dataset having target variable as income this informs whether an individual exceeds $50K/yr income or not. The dataset includes 48842 records with 14 attributes for each individual, some of these attributes are age, occupation, education, race, sex, marital-status, native-country, hours-per-week etc. with age and race as the sensitive attributes.

6.2 Synthetic Datasets

NG: we need to discuss in detail the dataset **Multiple Sensitive Attributes**: We notice that the synthetic data generated in [30] only has one sensitive attribute whereas we need to test our algorithms for multiple sensitive attributes. To that end we generate data using a Python function for generating synthetic datasets called `sklearn.datasets.make_classification`. with `n_samples`=2000, `n_features`=10, `n_informative`=6, `n_classes`=2 and `n_redundant`=4. We use the last five features as sensitive attributes, converting them to 0/1 values based on the condition $(>= 0)/(< 0)$. This data has non-zero disparate impact for all the five features and we can mitigate it using our algorithms.

6.3 Baselines

We compare the performance of our algorithms with four baseline algorithms namely with Zafar et al. [30], Agarwal et al. [1], Madras et al. [21] and Padala et al. [31]. We describe this comparison in the following:

NG: why is LP-1 coming into picture MP: LP1 now removed from the table

1. **Agarwal et al.**: A post-processing algorithm that reduces the fair classification problem to a sequence of cost-sensitive classification problems.
2. **Zafar et al.**: An in-processing algorithm that codes fairness constraints in the optimization problem of an SVM/LR.
3. **Madras et al.**: pre-processing algorithm that is based on representation learning and data transformation.
4. **Padala et al.**: An in-processing algorithm that encodes fairness constraints in the loss function of a deep neural net.

NG: best is to tell one-two lines about each baselines

Method	Type	DI/DP	Wgt.	Poly. Sens.	Multi. Sens..
LP-2	Post	✓	✓	✓	✓
LP-3	Post	✓	✓	✓	✓
Zafar et al [30]	In	✓	✗	✓	✓
Padala et al. [24]	In	✓	✗	✓	✓
Agarwal et al [1]	Post	✓	✗	✓	✓
Madras et al [21]	Pre	✓	✗	✗	✗

7 Experimental Evaluation

NG: Present a result without accuracy for each data at various β, only showing at what , the result become feasible Table where rows are data set and colums are various β and cells are value of ϵ In this section we demonstrate the performance of our frameworks for mitigating disparate impact and we compare them with existing frameworks. We have performed our experiment on real world dataset as well as on synthetic dataset as mentioned in Section 7.

Reproducibility: All the codes and datasets are available at https://github.com/subhampokhriyal94/fairness_using_discripancy_min

Adult Dataset: For the Adult dataset we have reproduced the training data as used by [30], by taking 10K samples in order to compare our results with his. We implement our framework for **LP-2** and plot the variation of positive class selection rates for sex (male, female) and race (white, black, asian, american-indian and other) in Figure 1. We consider two values of β namely 0.16 and 0.17 (which is roughly an average of all the positive class selection rates) and $\epsilon = 0.05$. We observe **LP-2** is feasible w.r.t these values of β, ϵ and we can mitigate the disparate impact these values upto an error of 2.4%. We then use our second framework merge the labelings obtained as above with that of the labels of the SVM using randomized approach. We observe that for $\beta = 0.17$, $\alpha = 0.5$ we get an error of 8% and the accuracy is 75%. If we compare this performance with that of [30], we notice that in his framework he mitigates the disparate impact to about 10% with an accuracy of 78%. Thus we almost match his bound for this choice of data. We compare the fairness vs. demographic parity trade-offs of our LPs w.r.t. four baseline algorithms, namely with that of [30], [1], [21] and [24]. The

DP	Padala's Accuracy	Our Accuracy	DP	Madras's Accuracy	Our Accuracy
0.04	79.8	80.0	0.08	82.1	81.7
0.02	81.3	80.7	0.14	82.5	83.6
0.047	79.0	80.6	0.17	83.6	84.6
0.05	78.4	80.9	0.12	81.8	82.9

Table 1: Test accuracies for the Adult Dataset of [24] and [21] for various DP values with sensitive attributes as 'sex'. Our parameters for **LP-2** are $\beta = 0.1, \delta = 0.8, \epsilon = .03$

DP	Zafar's Accuracy	Our Accuracy	DP	Agarwal's Accuracy	Our Accuracy
0.17	84.7	84.7	0.12	82.9	83.0
0.15	83.2	83.9	0.115	82.6	82.7
0.10	81.8	82.6	0.11	82.0	82.4
0.05	NA	81.0	0.09	81.9	82.0
0.01	NA	79.5			

Table 2: Test accuracies for the Adult Dataset of [30] and [1] for various DP values with sensitive attributes as 'sex' and 'race'. Our parameters for **LP-2** are $\beta = 0.1, \delta = 0.8, \epsilon = .03$

results are described below.

NG: What is disparate impact??

We are defining DI in terms of percentage as max-min: (acceptance rate of all values of sensitive features and writing it as percentage.)

Propublica Dataset ProPublica consists of two datasets violent and non violent. We again run **LP-2** and show similar results in Figure 4. For ProPublica violent we take β as 0.3 and 0.1; $\epsilon = 0.07$ and 0.14. We get minimum disparate impacts (always at $\alpha = 1$) of 3.5% and 7% respectively for these two cases. Applying the randomized approach we observe that for the former case we get a disparate impact of 11% with accuracy as 74% for $\alpha = 0.55$ and for the latter case, at $\alpha = 0.55$ we get a disparate impact of 9.8% with accuracy as 79%. For ProPublica described in Figure 3, non-violent we take β as 0.5 and 0.25 and $\epsilon = 0.01$. We get minimum disparate impacts of 5% and 4% respectively. Applying the randomized approach we observe that for the former case we get a disparate impact of 19% with accuracy as 60.75% for $\alpha = 0.5$ and for the latter case, at $\alpha = 0.55$ we get a disparate impact of 15.6% with accuracy as 60.61%.

Synthetic Data In the case of Synthetic data generated as mentioned in Section 7, we have 5 sensitive attributes each attribute consisting of binary values. We plot the variation of positive class selection rates of each of the 10 values in Figure 2. We take β as 0.5 and 0.4 and $\epsilon = 0.05$. We get minimum disparate impacts of 0.6% and 2.5% respectively. Applying the randomized approach we observe that for the former case we get a disparate impact of 19% with accuracy as 60.6% for $\alpha = 0.75$ and for the latter case, at $\alpha = 0.75$ we get a disparate impact of 17.3% with accuracy as 63%.

Feasibility of the LPs

An important question that needs to be answered while working with mathematical programming paradigms is to understand the conditions under which the program is feasible. In-

deed, as it turns out the linear program **LP-2** is infeasible for $\epsilon = 0$ and certain values of β_j for Adult dataset. But we can increase the value of ϵ to ensure feasibility. More, formally, we give a sufficient condition on the value of ϵ that ensures feasibility of the linear program. Let,

$$\epsilon_0 = (\max_j \beta_j) - (\min_j \beta_j)$$

From our experimental evaluation we have observed that if we choose $\epsilon = \epsilon_0$, then we always find a solution for the LP. Although we don't provide a mathematical proof of this statement but the intuitive reason for it is that we tend to see feasibility when $\beta_j = \beta$ and the number of constraints isn't too large. This condition is satisfied if we take $\epsilon = \epsilon_0$. To be realistic we would expect the user to provide $\beta_j's$ whose ϵ_0 is small say within 10%.

We observe that although in all the figures, the positive class selection rate eventually converges to the desired values (β), the convergence doesn't happen via a smooth function and we see jitter in their behavior (which we have smoothened using Savitsky-Goley filter). This behavior can be explained because of the use of randomization in the merging procedure. It is not clear if randomization is necessary for merging the colorings but it allows for both the colorings to "entangle" well.

8 Conclusion

In this paper we have proposed notions of fairness that are based on solving the well known problem of combinatorial discrepancy in theoretical computer science. We have used this notion along with tools from optimization like linear programming to develop a new framework for mitigating disparate impact which also considers a weighted version of the problem (using $\beta_j's$). We show that our approach solves important issues like fairness gerrymandering problem (to a certain extent) and data size blow-up for multiple polyvalent sensitive attributes [30] without compromising on the accuracy of the final classification. We also notice that our paper is the first paper to use the notion of combinatorial discrepancy in the field of fairness and we believe that this notion will help in developing a theory of algorithmic fairness in the future.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

[2] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 3–10. IEEE, 2010.

[3] Yahav Bechavod and Katrina Ligett. Learning fair classifiers: A regularization-inspired approach. *arXiv preprint arXiv:1707.00044*, pages 1733–1782, 2017.

[4] József Beck and Tibor Fiala. "integer-making" theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

[5] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[6] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

[7] Moses Charikar, Alantha Newman, and Aleksandar Nikolov. Tight hardness results for minimizing discrepancy. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1607–1614. SIAM, 2011.

[8] Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.

[9] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[10] JG Corput. van der: Verteilungsfunktionen. In *Proc. Ned. Akad. v. Wet*, volume 38, pages 813–821, 1935.

[11] Benjamin Doerr and Anand Srivastav. Multicolour discrepancies. *Combinatorics Probability and Computing*, 12(4):365–399, 2003.

[12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[13] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[14] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[15] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, pages 1–6. IEEE, 2009.

[16] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[17] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572, 2018.

[18] Shreeharsh Kelkar. The elite's last stand: negotiating toughness and fairness in the iit-jee, 1990–2005, 2013.

[19] Kasper Green Larsen. Constructive discrepancy minimization with hereditary l2 guarantees. *arXiv preprint arXiv:1711.02860*, 2017.

Disparate Impact	$\beta = 0.1$		$\beta = 0.2$		$\beta = 0.25$		$\beta = 0.3$	
	$\delta = 0.9$, $\epsilon = .18$	$\delta = 0.8$, $\epsilon = .3$	$\delta = 0.9$, $\epsilon = .08$	$\delta = 0.8$, $\epsilon = .01$	$\delta = 0.85$, $\epsilon = .02$	$\delta = 0.8$, $\epsilon = .02$	$\delta = 0.75$, $\epsilon = .01$	$\delta = 0.7$, $\epsilon = .01$
0.17	84.7	84.7	84.7	84.7	84.7	84.7	84.7	84.7
0.15	84.0	83.9	83.8	84.2	83.2	83.9	82.7	83.8
0.10	82.4	82.6	82.4	82.6	81.3	81.9	79.9	79.6
0.05	NA	81.0	80.9	81.0	79.9	79.5	76.9	76.5
0.02	NA	79.7	NA	79.7	79.0	78.3	75.7	75.7
< 0.01	NA	NA	NA	79.5	78.6	77.8	74.7	75.1

Table 3: For the Adult Data :Variation of accuracy w.r.t. Disparate Impact obtained by tuning parameters: β(acceptance rate), ϵ and δ.

Datasets (Sensitive Attrb.)	$\beta = 0.1$	$\beta = 0.15$	$\beta = 0.2$	$\beta = 0.25$	$\beta = 0.3$
Adult (Gender, Race)	0.004	0.0035	0.0025	0.0045	0.0025
Compass (Gender, Race)					
Bank (Age, Marital Status)	0.0004	0.0004	0.0005	0.0006	0.0007
German (Gender, Age)	0.02	0.025	0.035	0.045	0.05
Crime ()					

Table 4: LP2 Algorithm Is Applied To multiple Dataset.

No. of Sensitive Attrb. (k)	$\beta = 0.1$	$\beta = 0.15$	$\beta = 0.2$	$\beta = 0.25$	$\beta = 0.3$
$k = 2$	0.0005	0.0006	0.0005	0.0007	0.0004
$k = 4$	0.0006	0.0009	0.0007	0.0007	0.0008
$k = 6$	0.0006	0.0009	0.0007	0.0006	0.0007
$k = 8$	0.0009	0.0009	0.0008	0.0008	0.001
$k = 10$	0.001	0.001	0.001	0.001	0.001

Table 5

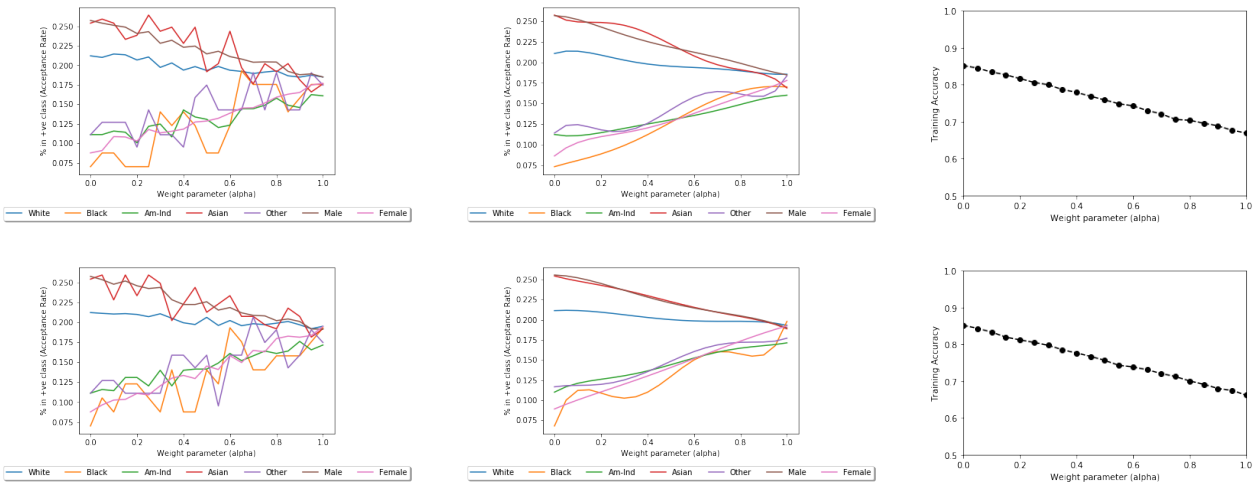

Figure 1: Variation of positive class selection rates across attributes for Adult dataset and Accuracy vs. α tradeoff using **LP-2** with $\beta = 0.16, \epsilon = 0.05$ (first row) and $\beta = 0.17, \epsilon = 0.05$ (second row). Second column represents smoothing of column 1 using Savitsky-Goley filter.

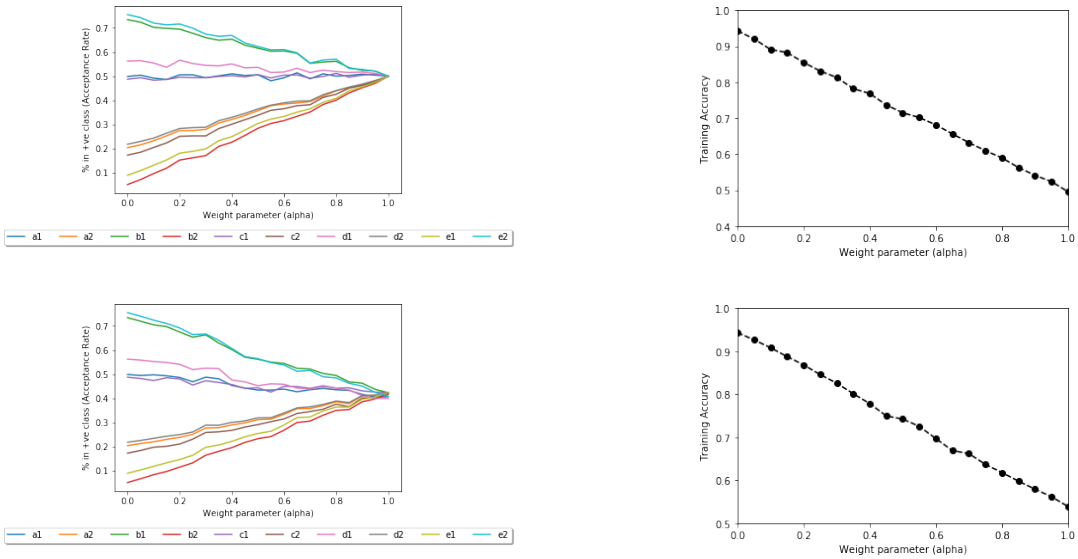

Figure 2: Variation of positive class selection rates across attributes for Synthetic dataset and Accuracy vs. α tradeoff using **LP-2** with $\beta = 0.5, \epsilon = 0.05$ (first row) and $\beta = 0.4, \epsilon = 0.05$ (second row).

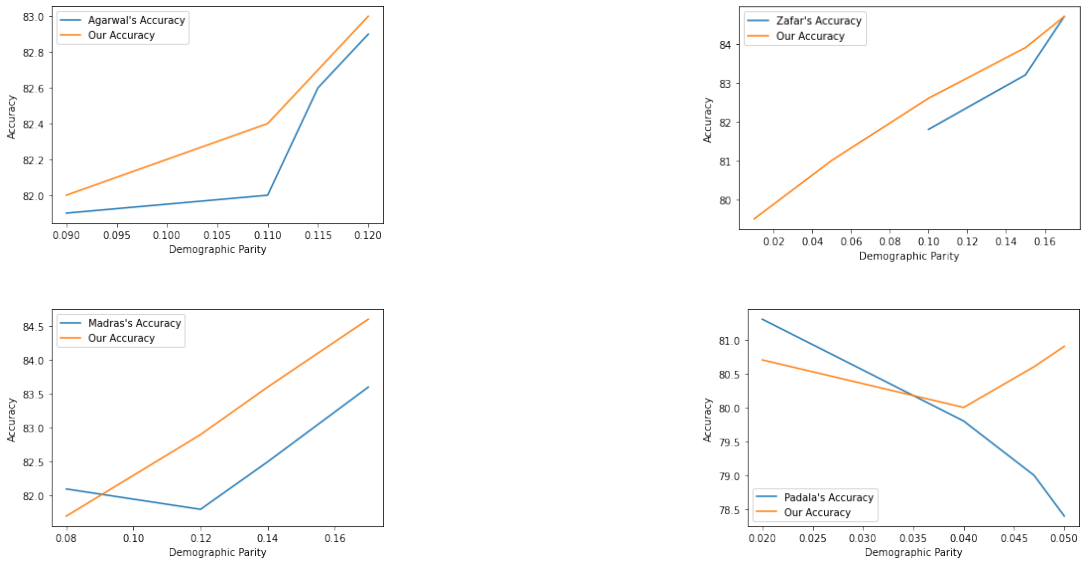

Figure 3

[20] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582, 2015.

[21] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 349–358, 2019.

[22] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[23] Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654, 2018.

[24] Manisha Padala and Sujit Gujar. Fnnc: Achieving fairness through neural networks. In *Proceedings of IJCAI*, pages 2277–2283, 2020.

[25] ProPublica.Org. Propublica risk assessment.

[26] Joel Spencer. Six standard deviations suffice. *Transactions of the American mathematical society*, 289(2):679–706, 1985.

[27] Terence Tao. The erdos discrepancy problem. *arXiv preprint arXiv:1509.05363*, 2015.

[28] UCI. Adult dataset.

[29] Yongkai Wu, Lu Zhang, and Xintao Wu. On convexity and bounds of fairness-aware classification. In *The World Wide Web Conference*, pages 3356–3362, 2019.

[30] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *J. Mach. Learn. Res.*, 20(75):1–42, 2019.

[31] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.