Fairness In Classification

Project Report
EE 380L: Data Mining

Marius Arvinte, Woody Austin, Saadallah Kassir, David Van Veen

January 28, 2019

Abstract

We present a convex fairness regularization penalty for use in classification tasks. This penalty both quantifies and reduces unfairness in predictions given a biased dataset. We define fairness as a function of one or more binary sensitive features. We first reproduce two penalties, one inducing fairness at an individual level and the other inducing group fairness, where variations from individuals within a sensitive class can be averaged out. We examine the performance of the fairness regularization term as a penalty added to a standard logistic regression model. We also propose three extensions for this work. First, we use combinations of multiple features as sensitive classes. Second, we extend the notion of fairness penalties on sensitive features to non-binary classes. Lastly, we look at the underlying notion that informs the fairness penalty and extend it to include a richer picture of what unfairness in data could mean. Our results highlight the tradeoff between fairness and model performance, which can be used as a measure of bias in the data.

1 Introduction

Machine learning is now ubiquitous in making decisions that have consequential impact on the lives of individuals in many domains such as credit, employment, education, and criminal sentencing [1, 4, 16, 19]. Unfortunately, due to bias in existing datasets and corresponding ground truth labels, these otherwise "unbiased" machine learning models perpetuate unfairness that already exists in the data. Such unfairness could affect many protected classes including gender, race, age, and disabilities.

The issue of biased machine learning models has led to real life examples of corporate embarrassment [14] which has driven corporations to work toward developing fair models. Similar issues of bias have led policymakers to take action as well. The European Parliament has recently adopted a set of regulations for the collection, storage, and use of personal information. These regulations, called the General Data Protection Regulation

(GDPR), will go into effect on May 25, 2018. GDPR will limit the authority that algorithms have to make decisions that "significantly affect" users. It also grants the "right to explanation," such that users can ask for an explanation of algorithmic decisions. This might even apply to seemingly inconsequential decisions such as why a user was shown a particular advertisement. These new regulations are predicted to cause a major disruption and have been described by some as a "Copernican Revolution" in data usage law [7].

Many perceive algorithms to be more fair in their decision making than humans because they are deterministic, unaffected by emotion, and have no agency [15]. The issue with this assumption is that learning algorithms are designed to learn existing statistical patterns in training data. Biased training data will lead to biased decisions; in some cases models can even amplify biases of the underlying data [12]. The goal of research into fair machine learning is to design models that make decisions which do not discriminate against a sensitive feature even given biased training data.

We note that there is an inherent tension between enforcing fairness over a biased dataset and achieving what would traditionally be considered high accuracy for a dataset. For example, if we expect that the ground truth with respect to salary is highly biased against women in a historical dataset, a fair classifier should predict that either more women are paid a higher salary, fewer men are paid a higher salary, or both. In any case, this fair prediction will *not* match the ground truth. Because of this, fairness is a tricky topic to quantify. There is not yet a universally accepted standard for what fairness in predictions means. We use the penalties that we implement in this paper as proxies for quantifiable fairness while trying to convey some qualifiable results via bar plots in Section 3. Even though these bar plots do not give a complete numerical picture of what is happening in the dataset, it gives the reader some intuition as to whether there is parity among predictions with respect to sensitive features.

1.1 The Failure of Naïve Approaches to Fairness

Intuitively, one might approach biased data by simply ignoring sensitive features. This idea of "fairness through unawareness" fails due to redundant encodings in the data. We can also think of these redundancies as correlation between sensitive features and and the rest of the data [18]. For example, race can be predicted quite accurately from basic personal information such as zip code and income [5].

Another naïve approach is to enforce probabilistic independence of sensitive features. For the case of a binary classifier, the probability of a decision $C \in \{0, 1\}$ is made independent of a sensitive feature S. In the case of gender as the sensitive feature, this strategy would enforce $Pr(C = 1) = Pr(C = 1|S = \text{male}) = Pr(C = 1|S = \text{female})$. There are issues with this seemingly rigorous approach [8]. Let's imagine a world in which all algorithms are legally obligated to enforce fairness through independence. This would yield very poor performance in model decisions; for example, consider the case of a cosmetics

company that is deciding whether or not to display a mascara advertisement to a particular user. If an algorithm must display such an ad with probability that is independent with respect to males and females, the lift metric of showing ads to a certain portion of users will decrease significantly. In other words, by ignoring the correlation between C and S, the model accuracy has decreased significantly. A second issue with this independence approach is that it permits "laziness". A model could predict very well for the dataset's majority class and have no incentive to predict well for the minority class. Essentially it can trade false positives for false negatives, which results in unfair decisions.

1.2 Our Approach

To explore the concept of fairness in machine learning, we will restrict ourselves to binary classification such that a model's output can be denoted as $C \in \{0, 1\}$. Our focus will be on logistic regression models because they are easily understood and widely used in many domains [6]. We compare standard or "vanilla" logistic regression to logistic regression with a fairness regularizer added to the loss function. This concept of a fairness regularizer has been explored by Berk et. al [2], whose methods we implement in addition to contributing our own novel notion of fairness.

As fairness is enforced upon the model, prediction accuracy will inherently decrease [8]. We aim to explore this fairness vs. accuracy trade-off for different notions of fairness, or "penalty types", and across different domains, or "sensitive features". These two different directions of extension are depicted in Table 1. The first way, through choice of penalty type, denotes the type of fairness we are enforcing: none, individual, group, or novel. The second way, through sensitive features, denotes which class we are enforcing fairness with respect to. This includes gender, race, and two different combinations of gender/race: pairwise summed and jointly combined. We discuss all of these at length in Section 2.

We make several novel contributions to this research area, as discussed in Section 2.4. The first is introducing a novel penalty type, which has the benefit of enforcing fairness for cases when true class labels are equal as well as when they are distinct.

Our second novel contribution is extending from binary sensitive features to M-ary sensitive features as well as to multiple sensitive features S with arbitrary arity.

We present analysis of results for the original formulation in [2], as well for our novel additions. We investigate the accuracy vs. fairness trade-off in different scenarios and show the consistency of our proposed improvements. We also look at the complexity of the new terms.

2 Theoretical Model

This section introduces the framework we have chosen to work with in order to quantify and enforce the fairness of a model, as per Berk et al. [2], as well as a series of novel extensions we propose. As mentioned in the introduction, *fairness* in itself does not have

a clear-cut definition. Rather, other works in the corpus of fairness literature (e.g. [11]) propose different ways of dealing with and measuring fairness.

The work that we have used to spark our exploration uses a convex and parallelizable regularization penalty. Other approaches use techniques like relying on estimating conditional probabilities. Unfortunately, the complexity of training the model using the technique that we selected grows asymptotically in each iteration from $\mathcal{O}(mn^2)$ to $\mathcal{O}(mn^3)$ where m is the number of features and n is the number of data samples. We address ways to deal with this increase in complexity in Section 2.5. A simple convex solver like the one we have chosen is still relatively simple compared to the machinery that goes along with Bayesian methods like MCMC or Variational Inference when applied to large-scale data [10].

2.1 Notation

All scalar quantities will be written in lowercase script (e.g. a). All vectors will be written in bold lowercase letters (e.g. $\mathbf{x}$). All matrices will be written in uppercase bold letters (e.g. $\mathbf{X}$). Indices will be written as subscripts and the typeface will change according to what the selection becomes. For example, the i^{th} entry of a vector $\mathbf{x}$ will be written. x_i and the j^{th} row of a matrix $\mathbf{Y}$ will be written $\mathbf{y}_j$. A special note: typically in scientific literature $\mathbf{x}_i$ would indicate the i^{th} column of $\mathbf{X}$, but since we are working with row vectors so frequently, we choose to let this represent the rows of $\mathbf{X}$ instead.

2.2 Logistic Regression

Let $\mathbf{X}$ denote the predictor matrix of size $n \times (k+1)$. Here k is the number of features, and we append an extra column of ones to the data matrix in order to account for a bias term in our weight vector. Let $\mathbf{y}$ denote the vector of binary class labels, and let μ_i denote the output probability $p(y_i = 1)$. $\boldsymbol{\beta}$ are the weights of our model of length $k+1$. Then we can write the log-odds ratio for sample x_i as:

$$\mu_i(\mathbf{x}_i) = \frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}. \tag{1}$$

The model is trained using the cross-entropy loss function [9]:

$$L_{CE} = -\sum_i y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i), \tag{2}$$

where the previous sum is taken across all training samples. We also apply an optional $l2$ regularization term to prevent overfitting, which is controlled by the parameter λ_2:

$$L = L_{CE} + \lambda_2 ||\boldsymbol{\beta}||_2^2. \tag{3}$$

2.3 Baseline Fairness Regularization Penalties

To gain an intuition about fairness in classification tasks, we implement two regularization penalties as in [2]. Given some *sensitive feature* S, we want to encourage our model to be fair with respect to the members of that feature set. For the first two formulations, assume that S is a binary feature.

Although S does not necessarily need to belong to $\mathbf{X}$ directly (meaning it can be supplied or defined separately), we assume that it is a part of $\mathbf{X}$ without loss of generality.

Based on S, we define two index sets Ω_+ and Ω_- such that Ω_+ is the set of all indices i such that $x_{i,s} = 1$. A similar definition follows for Ω_-.

For compactness, let $h_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i$. Now we can define the following two fairness regularization penalties:

1. The *individual* fairness regularizer:

$$L_{f,i} = \frac{1}{|\Omega_+||\Omega_-|} \sum_{\substack{(\mathbf{x}_i, y_i)\in\Omega_+ \\ (\mathbf{x}_j, y_j)\in\Omega_-}} d(y_i, y_j)(h_i - h_j)^2. \tag{4}$$

2. The *group* fairness regularizer:

$$L_{f,g} = \left[\frac{1}{|\Omega_+||\Omega_-|} \sum_{\substack{(\mathbf{x}_i, y_i)\in\Omega_+ \\ (\mathbf{x}_j, y_j)\in\Omega_-}} d(y_i, y_j)(h_i - h_j) \right]^2. \tag{5}$$

In both cases $|\Omega_i|$ is the cardinality of the set Ω_i. $d(y_i, y_j)$ is a non-negative function that is inversely proportional to the distance $|y_i - y_j|$. For binary classification problems, we choose $d(y_i, y_j)$ as the indicator function.

$$d(y_i, y_j) = \mathbb{1}[y_i = y_j]. \tag{6}$$

Note that we choose to use a different notation than in [2] (specifically the Ω_i set notation) because we find that the notation originally provided in that paper does not allow for enough precision in the definition of the penalty terms.

We now take a detailed look at Equations 4 and 5. Notice that, fundamentally, both expressions penalize the model when samples that have different members within a sensitive class have the same class label ($y_i = y_j$) but very different predictions.

The individual fairness regularization penalty applies this idea to all possible pairs of samples with different features within S. This discourages the model from producing different predictions as the regularization strength increases. Compared to the group penalty, this penalty is more aggressive. This is because the group penalty allows the errors between similar members within a sensitive feature to cancel each other out. Group fairness is a notion of average fairness within a sensitive feature rather than absolute fairness per sample.

Once we choose the fairness penalty to use, we weight it by λ_f and add it to our loss function in addition to the existing $l2$ penalty. The final expression of the cost function is:

$$L = L_{CE} + \lambda_f L_f + \lambda_2 ||\boldsymbol{\beta}||_2^2. \tag{7}$$

Finally, we note that both penalties presented in Equations 4 and 5 are convex and are therefore amenable to any form of gradient descent optimization when minimizing the loss function.

2.4 Novel Extensions

We present two paths upon which we've built novel ideas inspired by the core regularization penalty terms presented in [2]. The first is developing a new regularization penalty along with a theory of the types of unfairness that can arise in a biased dataset. The second involves extensions from binary sensitive features to M-ary sensitive features as well as to multiple sensitive features S with arbitrary arity. These two types of extensions are orthogonal concepts to each other and can be combined. These concepts are listed in Table 1, and we've bolded our novel contributions in this work.

Penalty Type	Sensitive Features
None	Gender
Individual	Race
Group	**Pairwise sum of Gender and Race**
Novel	**Jointly combined Gender and Race**

Table 1: Explanation of novel contributions. Bold entries are those that we have contributed in this work.

2.4.1 Improving the Existing Regularization Terms

Given two samples $\mathbf{x}_i, \mathbf{x}_j$ from different member classes Ω_1, Ω_2 within a sensitive feature, then there are two types of unfairness that can arise. The first type is the target of the baseline formulations presented in Section 2.3. Recall that $h_i = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i$. We can think of h_i as a proxy for the predicted probability given sample $\mathbf{x}_i$. Given that $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same true class label (i.e. $y_i = y_j$), we would expect that the difference between the predicted probabilities $(h_i - h_j)^2$ will be small. As this distance grows, so too do the existing penalty terms.

The second type of unfairness occurs when $\mathbf{x}_i$ and $\mathbf{x}_j$ have *different* true class labels (i.e. $y_i \neq y_j$). In this case, we would expect that the difference between the predicted probabilities $(h_i - h_j)^2$ should be relatively large. As this distance grows, the penalty value should decrease. With this intuition, we define a new fairness penalty that encourages both types of fairness:

$$L_{f,dual} = \frac{1}{|\Omega_1||\Omega_2|} \sum_{\substack{(\mathbf{x}_i,y_i)\in\Omega_1 \\ (\mathbf{x}_j,y_j)\in\Omega_2}} \left[\mathbb{1}[y_i = y_j]\,(h_i - h_j)^2 + \mathbb{1}[y_i \neq y_j] e^{-(h_i-h_j)^2}\right]. \tag{8}$$

The two types of unfairness are visually displayed in Figure 1. We can imagine that our fairness penalty is softly constraining our predictions to be closer to the fair diagonal of that figure.

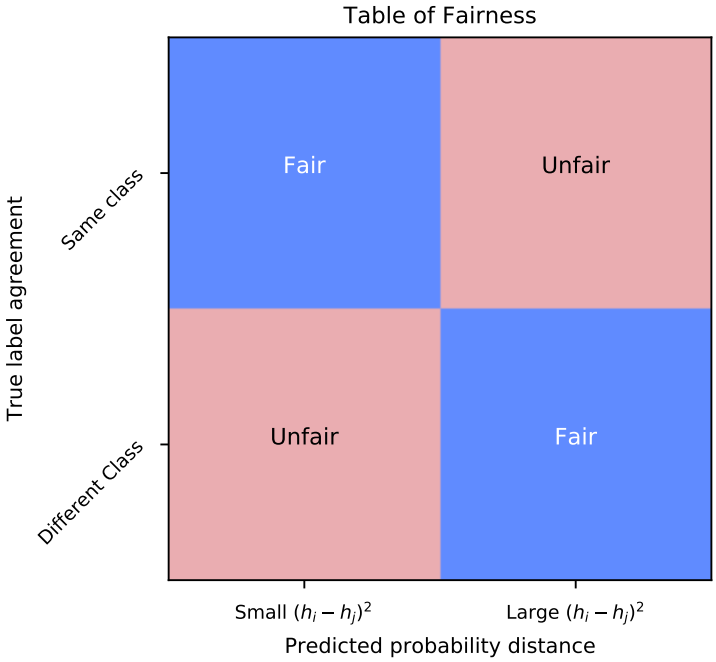

Figure 1: There are two types of fairness that can be imposed in a convex penalty. The first type is when, between two distinct protected class samples, the true class is the same. In this case, we desire that the distance between predicted probabilities for each sample should be small. The baseline penalties impose this type of fairness. The second case occurs when the true class labels are different. We would expect that the distance between predicted probabilities for each sample should be relatively large. Our new extension to the penalty incorporates this type of fairness.

2.4.2 Pairwise and Joint Fairness Regularization Terms

We now propose and motivate two more novel extensions to the fairness regularization penalty presented in Section 2.3. These new extensions also apply to the novel formulation for fairness presented about in Section 2.4.1. In fact, we can treat the extensions in this section as meta-learning strategies for any of those three fairness penalties. For brevity, we will restrict the expressions of our new penalties to those of the group fairness regularization penalty in Equation 5, but each new penalty can be trivially extended to the individual fairness and novel fairness versions as well.

One fundamental limitation of Berk et al. [2] is that the regularization penalties are defined only for a single binary sensitive feature S with no suggestions on how to extend the expressions to non-binary or multiple feature(s). An obvious limitation of only handling binary features is apparent in racial discrimination. Given the previously introduced penalties, we can only enforce fairness for two races rather than the myriad of ethnicities that exist in the real world.

Another type of extension to this method is when one desires a model that is fair with respect to two (or more) binary sensitive features, for example gender *and* race. Our first, simple, approach in this case is to use Equation 5 for each of the features:

$$L = L_{CE} + \lambda_f L_f + \lambda'_f L'_f + \lambda_2 ||\boldsymbol{\beta}||_2^2, \tag{9}$$

where L_f and L'_f are the penalties for distinct sensitive features. However, this regularizer only enforces the fairness with respect to S and S' individually and does not capture an explicit notion of **joint fairness**.

Consider a collection of binary sensitive features labeled S_i. We can define index sets of each combination of features. For example in the case of two binary features $\{S, S'\}$, we can define the following:

$$i \in \Omega_j, j = \begin{cases} 0 & \text{if } (x_{i,s}, x_{i,s'}) = (0,0), \\ 1 & \text{if } (x_{i,s}, x_{i,s'}) = (0,1), \\ 2 & \text{if } (x_{i,s}, x_{i,s'}) = (1,0), \\ 3 & \text{if } (x_{i,s}, x_{i,s'}) = (1,1). \end{cases} \tag{10}$$

Since categorical variables like race are typically dummy coded, the Ω_j values from Equation 10 are an appropriate definition for a ternary sensitive feature if we only ignore the Ω_3 case, which will never happen in such an encoding. We can now introduce two novel extensions to the regularization penalties inspired by [2] that enforce some notion of multiple fairness.

2.4.2.1 Pairwise Fairness Regularization Penalty

The following penalty enforces pairwise fairness across all possible pairs of sets of Ω_j by adding one based on Equation 5 for each such pair. As an example, in the case of a ternary S_j with the resulting set of index sets $\{\Omega_1, \Omega_2, \Omega_3\}$, the expression of the proposed regularization term is:

$$L_{f,pairwise} = \left[\frac{1}{|\Omega_1||\Omega_2|}\sum_{\substack{(\mathbf{x}_i,y_i)\in\Omega_1\\(\mathbf{x}_j,y_j)\in\Omega_2}} d(y_i,y_j)(h_i-h_j)\right]^2 + \left[\frac{1}{|\Omega_2||\Omega_3|}\sum_{\substack{(\mathbf{x}_j,y_j)\in\Omega_2\\(\mathbf{x}_k,y_k)\in\Omega_3}} d(y_j,y_k)(h_j-h_k)\right]^2 + \left[\frac{1}{|\Omega_3||\Omega_1|}\sum_{\substack{(\mathbf{x}_i,y_i)\in\Omega_3\\(\mathbf{x}_k,y_k)\in\Omega_1}} d(y_i,y_k)(h_i-h_k)\right]^2. \tag{11}$$

Obviously, this notion can be extended for any M-ary sensitive feature S. Given M possible values for a feature S, the regularization penalty will contain $\binom{M}{2} = \frac{M(M-1)}{2}$ sums that need to be computed (a detailed discussion on complexity will be done in a later section). Thus, this formulation penalizes any significantly different outputs between members of a class and all others. This is very similar to one-versus-one classifiers in classification literature [10].

Also note that this expression can be used in the case when S is binary *and* when we have a collection of binary or M-ary features. Thus, this is a direct generalization of the work in [2]. As mentioned before, the same type of expression can be derived for the individual type of fairness regularization penalty.

Note that when there is a collection of features, we will have to slightly augment our notation to handle multiple sets of indices corresponding to the set of sensitive features. Something like $\Omega_1^{(2)}$ could represent the first index set of the second sensitive feature. We will see such an example in the following section.

2.4.2.2 Joint Fairness Regularization Penalty

We propose a second novel regularization term that measures the weighted prediction error with respect to more than one sensitive feature S. In contrast with the pairwise regularization term that is applied for each feature separately, the joint term first constructs the cartesian product of all sensitive features, and then applies the pairwise fairness principle.

Consider the case of two sensitive features S_1 and S_2, each of arity K_1 and K_2 respectively and their corresponding sets of index sets:

$$\begin{aligned}\Omega^{(1)} &= \{\Omega_1^{(1)}, \Omega_2^{(1)}, \ldots, \Omega_{K_1}^{(1)}\},\\ \Omega^{(2)} &= \{\Omega_1^{(2)}, \Omega_2^{(2)}, \ldots, \Omega_{K_2}^{(2)}\}.\end{aligned} \tag{12}$$

We construct the joint set of index sets Ω as the Cartesian product

$$\Omega = \Omega_1 \times \Omega_2, \tag{13}$$

and we define the joint regularizer by enforcing fairness between any two sets Ω_k and Ω_l in Ω:

$$L_{f,joint} = \sum_{k,l} \left[\frac{1}{|\Omega_k||\Omega_l|} \sum_{\substack{(\mathbf{x}_i,y_i)\in\Omega_k \\ (\mathbf{x}_j,y_j)\in\Omega_l}} d(y_i,y_j)(h_i - h_j) \right]^2 . \tag{14}$$

The previous equations can be extended for the case when we are dealing with an arbitrary number of sensitive features, each with arbitrary arity; we simply construct the n-fold Cartesian product Ω and apply Equation 14 directly.

Furthermore, note that the previous equation becomes identical to Equation 11 when dealing with a single sensitive feature. When dealing with multiple sensitive features the joint regularization term enforces the fairness in a stronger way by considering all possible interactions between features. In contrast to the pairwise sum extensions which imposes a one-versus-one strategy for fairness, this joint extensions is more akin to one-versus-all classifiers in the literature [10].

Finally, both newly introduced expressions reduce to the baseline term in Equation 5 when dealing with a single, binary S, thus our work can be viewed as a generalization of [2]. As mentioned before, similar definitions can be introduced for the individual and novel fairness regularization terms, but we omit them because there is no new theoretical insight to be gained from doing so.

2.5 Design and Implementation Choices

We implement logistic regression and each of the above fairness penalties using Python and the PyTorch library [17]. We chose to use PyTorch because it allows fast prototyping of models with features like automatic differentiation, data loading utilities, highly performant optimization methods, built-in loss functions, and GPU support.

We design our classification method to include optional minibatching, online visualization, and validation sets. The design of the penalty method takes advantage of the remarkable overlap between each of the formulations given in Section 2.4. For example, the indicator functions for equality of class labels appear in all formulations. We take advantage of spatial locality and, optionally, the power of GPUs by batching the differences as broadcast subtractions between the class labels (and h_i) in each index set. The remaining calculations are simply element-wise multiplications and sums of resulting matrices, both of which are highly optimized in the CUDA kernels that PyTorch employs for its GPU operations.

We quickly realized that it is prohibitively computationally expensive to compare fairness to every sample in the entire epoch. Since we use Adam [13] with minibatching as our

optimization method of choice, we chose to calculate fairness penalties per minibatch and sum them together over an epoch. One can think of this as sampling the formulations given above, just as minibatched SGD [3] can be thought of as randomly sampling traditional Gradient Descent. In practice, we found that a slightly larger than typical minibatch size gives a more representative picture of fairness. We settled on minibatch sizes of 512 samples. Empirically, we have found that doing within minibatch fairness calculations yields results (in accuracy and fairness metrics) comparable to full fairness calculations.

The result of all of these design choices is a major speedup of these asymptotically complex regularization penalties.

3 Results

3.1 Dataset

We work with the *Census Income* Dataset from UCI, which is also referred as the *Adult* dataset. Each sample corresponds to information related to a specific employee. The columns types are characterized in Table 2. The dataset consists of more than 48,000 entries which we broke into a $\frac{2}{3}$ train/test split. We choose to work with the *Adult* dataset because it contains several features that are typically considered protected, including gender, ethnicity, and nationality.

Most of the features have very descriptive names, with the exception of two. *WorkClass* represents the classification of an employee such as federal, private, *etc.* *Fnlwgt* is a metric based on demographic characteristics such as race, age, and sex. The closer the weights are, the more "similar" the individuals are based on these features. This particular feature is highly correlated with all other features and we remove it during preprocessing. For our classification task, we discretize income into a binary class such that the threshold between low and high income is set to be \$50K a year (i.e. $\leq$ \$50k is a negative example and $>$ \$50k is a positive example).

3.2 Data Preprocessing

The raw data contains many categorical features and irrelevant information. Before working with it, we first preprocess the dataset. Our preprocessing transforms categorical variables into numerical quantities without introducing any erroneous linear relationships within categories.

We describe our preprocessing procedure below:

1. Use heuristics and domain knowledge to drop irrelevant and redundant features. For this particular dataset, we drop the *Fnlwgt* and *Education* features.

2. Visualize the dataset to understand how the data is distributed within each feature.

Feature name	**Type**
Age	Continuous
Workclass	Categorical – 8 classes
Fnlwgt	Continuous
Education	Categorical – 16 classes
Education-num	Continuous
Marital-status	Categorical – 7 classes
Occupation	Categorical – 14 classes
Relationship	Categorical – 6 classes
Race	Categorical – 5 classes
Sex	Categorical – 2 classes
Capital-gain	Continuous
Capital-loss	Continuous
Hours-per-week	Continuous
Native-Country	Categorical – 41 classes
Income	Categorical – 2 classes

Table 2: Raw feature names and types for the *Adult* dataset. Because of the high number of categorical features, we reduced the dimensionality of the data via feature engineering. In addition to both dummy coding categorical features and compressing features, inspection of the cross-correlation matrix reveals that some features are redundant and can be eliminated completely.

3. Merge categories based on the insight gained from visualizing the data. For this dataset, categorical features such as *Country*, *Race*, and *Workclass* are compressed into binary features by merging the least frequent classes together into a single class. For example *Country* becomes *US/Non-US*. This step significantly prunes the feature space in our new encoding. Figure 2 shows the unbalanced distribution of the data among the *Country* feature, where most of the employees come from the US. By itself, compressing this feature into a binary one allows us to reduce the dummy coded feature space size by 39.

4. Use dummy coding to convert the entries of the (now compressed) categorical features.

5. Center and normalize the continuous entries so that they are approximately normally distributed with mean centered around zero and variance of one.

6. Run Lasso to detect and remove features with low predictive power. For this step, we determine the hyperparameter for the $l1$ penalty using cross-validation.

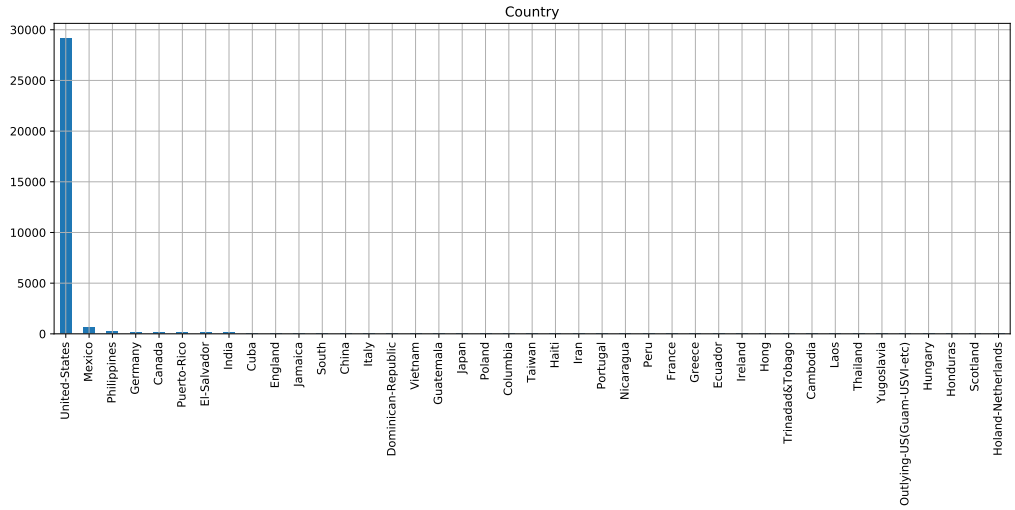

Figure 2: Histogram of the *Country* feature in the *Adult* dataset. This is an example of a categorical feature that is heavily skewed and can be compressed to a much lower dimensional space (e.g. binary or ternary) without noticeable performance loss. We choose a binary *US/Non-US* relationship; another option is a ternary relationship where the members were *US/Mexico/Other* as well.

7. Verify that the remaining features are reasonably uncorrelated among each other and correlated with the target feature by examining the correlation matrix shown in Figure 3.

Using this procedure, we encode all useful categorical and continuous information into only 27 features. With only dummy coding and none of our preprocessing, the feature space would encompass over 100 unique features.

3.3 Empirical Results

3.3.1 Logistic Regression with Classic Fairness Regularization

We use standard logistic regression with no fairness regularization as a baseline for all of our experiments. We define overall accuracy to be:

$$OA = \frac{1}{n}\sum_{i}^{n}(\hat{y}_i - y_i)^2, \tag{15}$$

where $\hat{y}_i$ is the prediction of a model and n is the number of samples. The overall accuracy for vanilla logistic regression on this dataset is 84.9%. We compare the baseline performance of the logistic regression model to the performance with the various fairness regularization penalties added. In Figure 4, we plot the classifier accuracy versus the regularization hyperparameter as it increases.

Note that for all experiments below, the results are the mean of three-fold cross validation on the training data that is then predicted using the same test data set. To be clear,

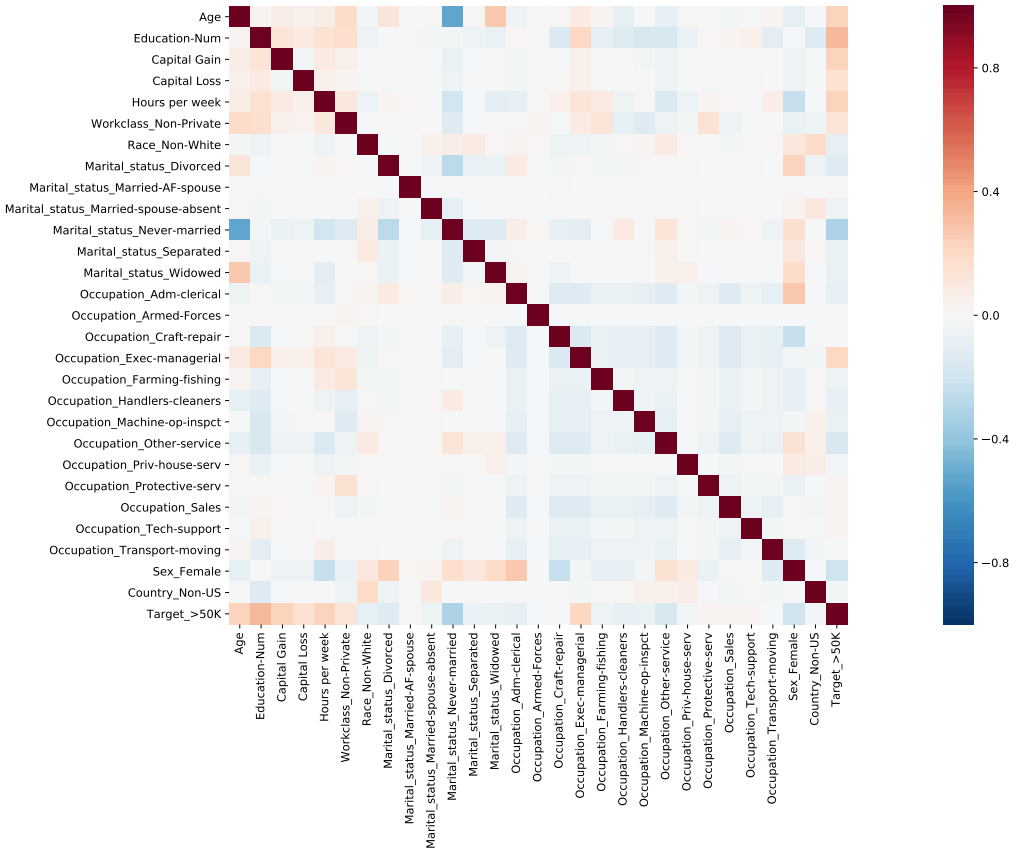

Figure 3: Cross-correlation matrix of the final 27 features after pre-processing. The high positive values on the diagonal and the lack of any noticeable correlation clusters indicate the final features are not correlated, hence reducing impact of the co-linear problem. *Age* and *Marital_status_Never-married* do have strong anti-correlation with each other, but neither of these features is strongly correlated with any other.

the hold-out sets in each cross validation fold are completely separate from the test data.

As we increase the fairness penalty, the accuracy of the classifier drops from 84.9% to 76% for this specific dataset. When the penalty is set to capture unfairness at the individual level, the model performance is affected at smaller values of λ_{fair} than in the group fairness scenario. This is due to the fact that the individual function is a harsher version of fairness than group where inequity can average out across all samples.

3.3.2 Logistic Regression with Novel Fairness Regularization

In addition to reproducing and analyzing results of the individual and group fairness regularization terms defined in [2], we evaluate and compare the performance of our novel regularization penalty we proposed and described in Section 2.4. The red curve in Figure 4 shows that for the same amount of penalty, the novel penalty has better accuracy than its individual fairness counterpart. It reaches a steady value of about 78% overall accuracy when λ_{fair} is large enough, while the individual fairness penalty plateaus at an

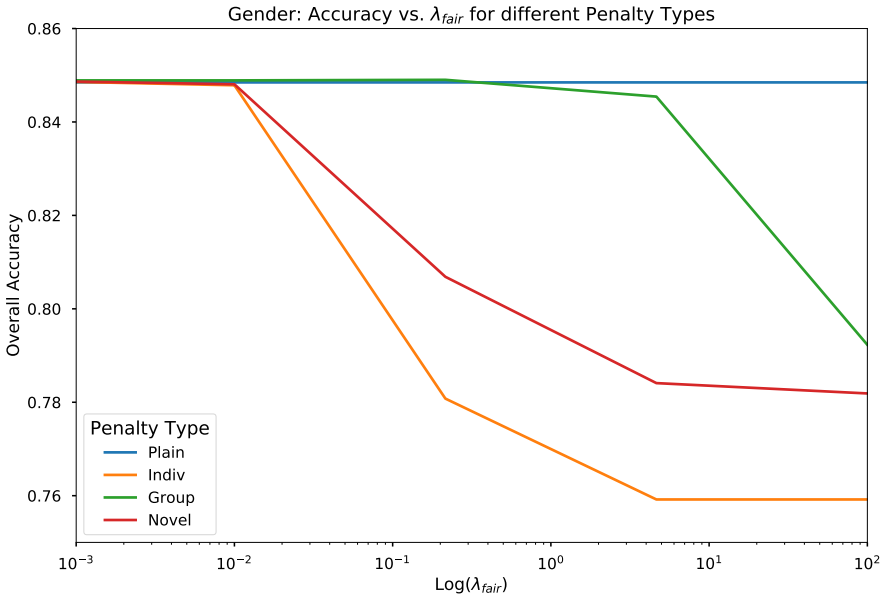

Figure 4: Classifier Performance as a function of λ_{fair}. This shows that as you increase the hyperparameter λ_{fair}, model accuracy decreases differently for different penalty types. We see roughly a 9% decrease in performance while using the individual penalty. Note that plain corresponds to no fairness penalty, so the model accuracy remains constant.

accuracy score of 76%.

Figure 5 shows that unfairness in the data is reduced as the weight of the fairness penalties increase. The higher the score on the y-axis, the more unfair the model is. Note that we use the individual fairness metric in this figure. Figure 6 shows the same results, but using the novel fairness metric. Comparing these two figures, one can deduce that although the novel fairness metric captures joint information from the gender and race features, one does not gain much more information compared to the individual fairness metric. Therefore, any of these two metrics are suitable to carry out further analysis.

The analysis of Figures 4, 5, and 6 show a clear trade-off between the classifier accuracy and the model fairness. Here again, we observe that the model becomes more fair for smaller values of λ_{fair} when the individual fairness penalty is used compared the group fairness penalty. We observe similar fairness scores between the individual and our novel fairness penalties. One can conclude from Figure 4 and 5 that for similar levels of individual fairness, the novel regularization term behaves better than the one based on individual fairness, as it leads to more accuracy in predicting the true class labels while maintaining higher overall accuracy. Upon further analysis, we notice that the individual and novel penalty types each seem to converge to a fairness score for a *lower* value of λ_{fair} (Figure 5) than for the convergence of accuracy in (Figure 4). This means that as λ_{fair} increases, at a certain point you are gaining marginal fairness while losing substantial accuracy. Thus it

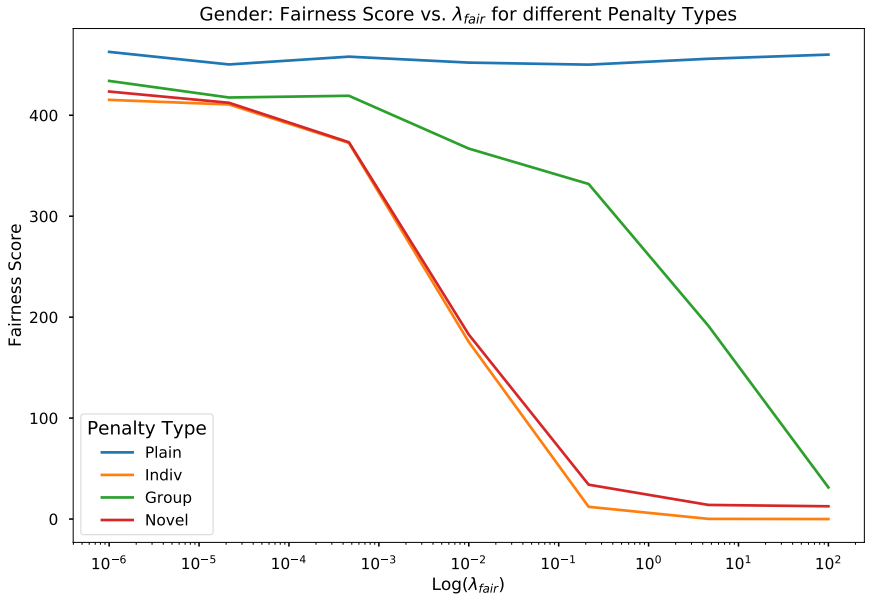

Figure 5: Individual Fairness Score as a function of $\log(\lambda_{\text{fair}})$, where a lower score indicates that the model is more fair. This shows that fairness score decreases differently for different penalty types. Plain corresponds to no fairness penalty, so the model fairness remains constant.

might be sensible to choose a value of $\lambda_{\text{fair}} = 2 \cdot 10^{-1}$ since the maximum fairness is almost achieved (Figure 5) while the model's predictions are still significantly more accurate than for higher values of λ_{fair} (Figure 4).

We show the effect of the fairness regularization on the dataset for male versus female members of the gender class as they relate to high/low salary prediction in Figure 7. The leftmost bars in the figure represent the percentage of males and females having a low income in the true class labels. The second set of bars shows the output of the vanilla logistic regression classifier. The leftmost and center bars look very similar due to the relatively good performance of the classifier. The rightmost set of bars represents the output of the classifier when regularized for fairness. We note that the performance of the the group and novel regularization penalties are identical, as observable comparing the third and fourth pairs of bars. The predictions of the fair regularization models based on the biased data seem to be more fair, at least with respect to the percentages of males and females in the low income class.

The bar plot in Figure 7 shows aggregated fairness among all the individuals within the male/female members, which can be too simple a visualization because it does not show the relationship among classes that are strongly correlated with the true class labels. Because of this, we show similar plots in Figure 8 that capture such correlated features, namely education (in years), with the income variable. The prediction errors are more

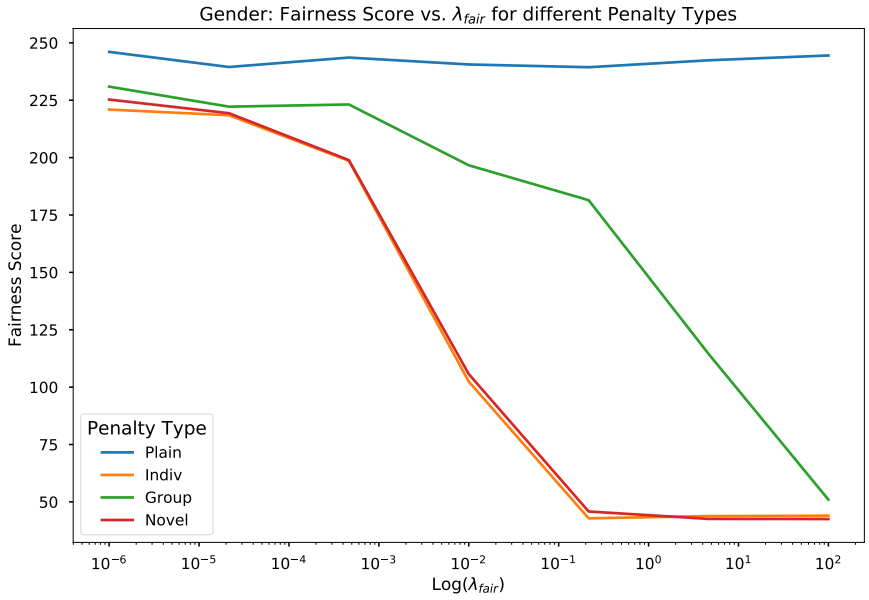

Figure 6: Novel Fairness Score as a function of $\log(\lambda_{\text{fair}})$, where a lower score indicates that the model is more fair. This shows that fairness score decreases differently for different penalty types. Plain corresponds to no fairness penalty, so the model fairness remains constant.

visible on this figure. In this figure, the comparison should be the difference between the light and dark bars. For the true class labels and the vanilla logistic regression predictions, one can see a large discrepancy between the light and dark bars of each color. Whereas, the "fairness gain" for the regularized predictions is noticeable. For clarity (to avoid too many bars on one plot), only the group fairness regularization output for a subset of the education levels has been displayed on the figure.

3.3.3 Logistic Regression with Multiple Sensitive Features

Before we analyze the numerical performance of the two novel extensions that allow us to control fairness across multiple sensitive features, we must note one important aspect. Previously we were able to measure the fairness with the individual regularization term (e.g. as in Figure 5), now we have no clear way of doing so. This is because the scale of the pairwise sum's unfairness score is much higher than the joint formulations unfairness score. This makes sense intuitively as the joint formulation is comparing unfairness across comparatively smaller combined similar members within a sensitive class. As a concrete example, the number of white males will be less than the number of males and the number of white samples. Likewise, the number of non-white females will be less than the number of females and the number of non-white samples. We will instead focus on the accuracy loss of the model as λ_{fair} increases.

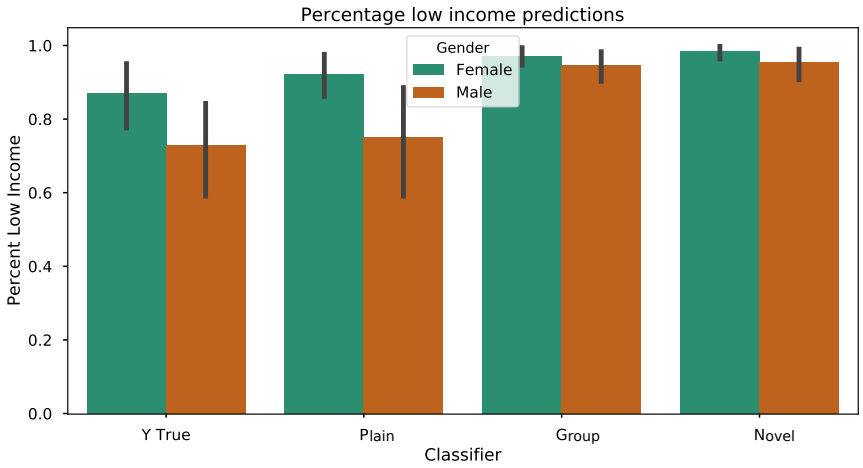

Figure 7: Bar plot denoting percentage of low income per gender. Y True represents the data set, while Plain represents classification with no fairness regularizer. When different types of fairness (Group, Novel) are implemented, we see the percentage of low income predictions are more similar for males and females, indicating the decisions are more fair.

Table 3.3.3 summarizes the accuracy loss incurred by our proposed terms for enforcing multiple and single fairness. The first two lines correspond to scenarios where the sensitive feature is binary. They serve as a baseline and show that the joint regularization term falls back to the expression in Equation 8.

The last two lines corresponds to scenarios where we enforce the model to be multiply fair with respect to race and gender. We can see that both methods exhibit the same accuracy vs. fairness trade-off, but surprisingly, the joint regularization term has a slower decay.

Sensitive Feature	$\lambda_{fair} = 0$	$\lambda_{fair} = 100$	% Accuracy Lost
Race	0.8485	0.7662	8.23%
Gender	0.8485	0.7819	6.66%
Pairwise-summed	0.8485	0.7592	8.93%
Jointly-combined	0.8485	0.8154	3.31%

Table 3: Model Accuracy across different (combinations of) sensitive features. The first two columns show the model accuracy without a fairness penalty, $\lambda_{fair} = 0$, and with a fairness penalty hyperparameter $\lambda_{fair} = 100$. The final column denotes how much accuracy is lost enforcing fairness for that sensitive feature. All results are reported with respect to our novel penalty type.

The unfairness score as a function of λ_{fair} is shown for the pairwise and joint regularization terms in Figure 9, where the scoring used is the one in the *Novel* expression extended as a pairwise sum of features (left y-axis label) and a joint combination of fea-

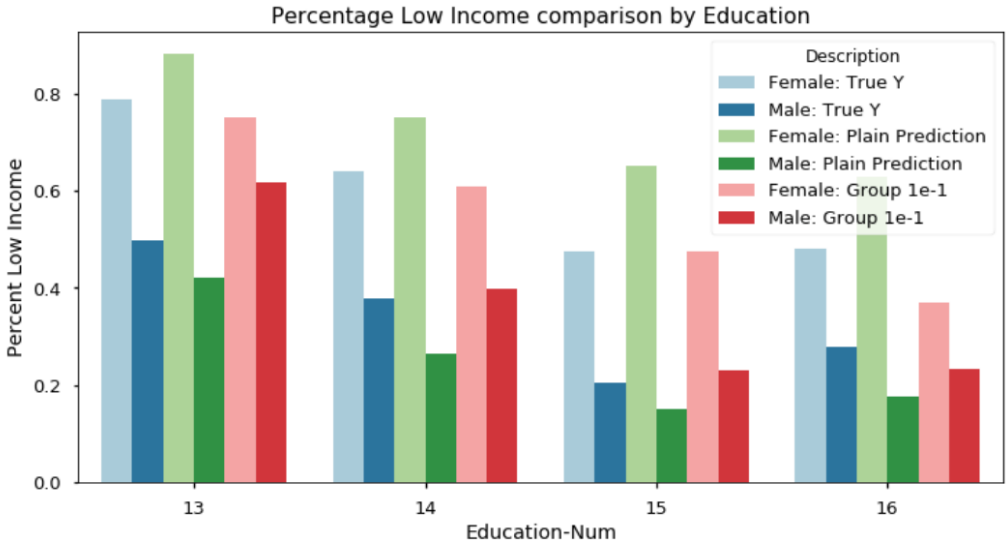

Figure 8: Predictions with and without the fairness classifier broken down by education level. We only use higher levels of education in the plot because they are the most illustrative of the wage gap between males and females. For each color, lighter shades represent the percentage of female samples that are predicted to be paid a lower wage and darker shades represent the percentage of male samples that are predicted to be paid a lower wage. Notice that the gap in the regularized predictions is visually less than that of the true and unregularized predictions.

tures (right y-axis label). Both terms exhibit a decay of unfairness when increasing λ_{fair}, but the order of magnitude is different, suggesting that the pairwise and joint formulations cannot be compared directly.

4 Key Findings

The main challenges encountered while working on the project were:

- Defining *fairness* and how to enforce it in a model. Extending this when trying to apply them to a non-binary or multiple sensitive features is not straightforward. Trivial extensions will not capture **jointly fair** models, while our proposed extensions do.

- Even though fairness regularization seems like a basic idea and the expressions are convex, there is an increase in complexity from $\mathcal{O}(N^2M)$ to $\mathcal{O}(N^3M)$ per epoch. For $N \gg 1000$ this quickly becomes prohibitive, thus batched and parallelized computation is imperative.

- The number of possible configurations (e.g. individual or group, pairwise or joint) is prohibitively expensive to simulate, analyze, and visualize exhaustively. One should

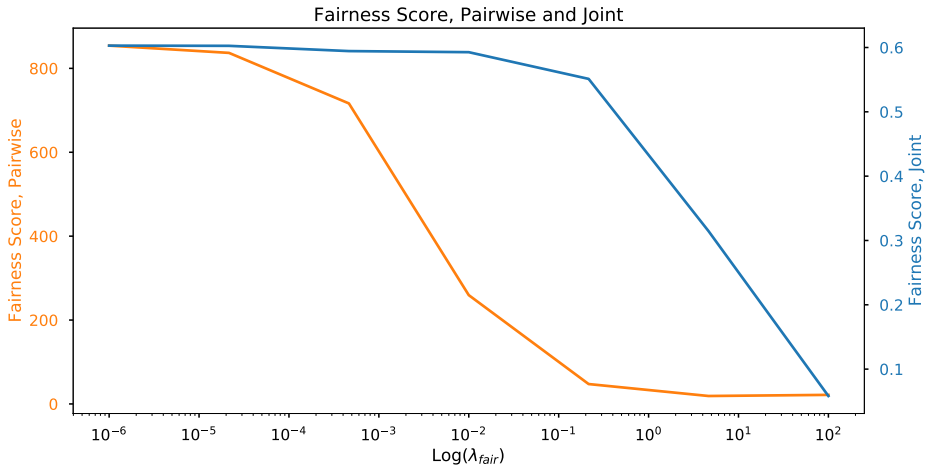

Figure 9: The unfairness score of the proposed pairwise and joint regularization terms in Equations 11 and 14 as a function of the regularizer strength λ_{fair}.

know in advance what type of fairness is required and in what way it should be enforced in a model.

4.1 Discussion on Complexity

We analyze the computational cost of the proposed improvements in Section 2.4 by looking at how their algorithmic asymptotic complexity per epoch scales with the number of training samples N and the size of the ensemble of protected features S.

We count the total number of terms of the form $(h_i - h_j)$ in each expression as a function of N and K, where K is the arity of the sensitive feature S. We also look at the asymptotic complexity as a function of N and the number of training features M. Note that the complexity is not increased in an asymptotic sense by the $d(y_i - y_j)$ function because it has the same number of operations as calculating $(h_i - h_j)$ and thus only increases the complexity by a factor of 2, which disappears in the asymptotic case.

Both expressions in Equations 11 and 14 involve a number of sums across all possible pairs in a set with the terms in each sum being proportional to the number of training points that belong to that set. Without loss of generality, we assume that the training points are equally split between the different values of the sensitive feature (or features, in the joint case) S. In fact, this serves as a worst-case upper bound on the number of summed terms and therefore is exactly the case that we should be analyzing for asymptotic complexity. The results are summarized in Table 4.

Remarkably, the asymptotic complexity of the multiple feature regularization terms remains the same. This is explained by the fact that, for a fixed N, even though the number of sums increases, each of them contains fewer terms with increasing K, thus the total number of terms per iteration is always proportional to N^2.

Regularizer	Summed terms per sample	Asymptotic complexity of one epoch
Classic [2]	$\frac{1}{4}MN^2$	$\mathcal{O}(N^3M)$
Pairwise	$\frac{K-1}{2K}MN^2$	$\mathcal{O}(N^3M)$
Joint (S_1 and S_2)	$\frac{K^2-1}{2K^2}MN^2$	$\mathcal{O}(N^3M)$

Table 4: Complexity evaluation of the classic and novel regularization terms in the worst case, when the data is equally split among the values of the sensitive feature S. The complexity for the joint term considers a scenario with two sensitive features of equal arity.

5 Conclusions

5.1 Results Summary

Our study gives intuition as to the effect of fairness regularization on classification problems using biased datasets. We conclude (somewhat obviously) that there is the trade-off between accuracy and fairness; i.e. the fairer we want our model to be, the less accurate it becomes. The magnitude of this trade-off varies between datasets. This effect is observable for all of the regularization penalties we study in this work. The trade off appears later for the group fairness penalty than individual fairness with respect to the penalty weight. Our novel extensions to the regularization penalties, which jointly capture unfairness in two or more sensitive features, behave similarly. This indicates that these formulations make sense because they follow similar trends to the established fairness regularization methods. We show that these new penalty functions lead to better prediction accuracy, while maintaining roughly similar fairness scores compared to the penalties proposed in Berk et al. [2].

5.2 Future Work

Based on the results we have obtained so far, there are several extensions that could complement our preliminary exploration of the subject, especially considering that the definition of fairness is not a clear-cut one.

One such extension is further validation of our work by simulating the performance of the different regularization terms on a new dataset. We could verify extreme scenarios by creating an artificial dataset and checking the behavior of the fairness-constrained model on such wildly biased inputs.

Another direction of research is to apply our proposed novel and joint regularization terms to linear regression (and other classification) models using appropriate $d(y_i - y_j)$

functions for suitable datasets. Note that this involves a reformulation of Equation 8, since we cannot use indicator functions anymore. Still, we believe that this extension should be straightforward and easy to reason about.

Finally, since we have seen that we cannot compare the two proposed terms for multiple fairness directly, it would be interesting to come up with a proper measure of fairness with respect to multiple, non-binary sensitive features. Such a measure of fairness should encompass inherent bias in the predicted labels and the predictor matrix $\mathbf{X}$ rather than specifically with respect to sensitive features S.

6 Code Repository

Our code repository is publicly hosted at `https://bitbucket.org/austinwn/ee380l_final/src`. Although we don't anticipate many updates after submission, a git tag has been created under the name "`final_submission`" to ensure that anyone examining this document can find the correct version of our code.

7 Azure Feedback to Instructor

While we appreciated being provided the option of a high-speed cloud computing platform, our group did not make any use of Azure for our project. Two of our members have access to machines with multiple high-end GPUs, and we found this a much preferred option. The main reason behind this choice is probably due to the familiarity with our own machines and not wanting to jump through the hoops required for Azure. So while the availability of cloud computing for the class was a nice option and was useful for the homework assignments, our group did not need it for the project itself.

References

[1] Anna Maria Barry-Jester, Ben Casselman, and Dana Goldstein. The new science of sentencing. *The Marshall Project, August*, 8, 2015.

[2] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

[3] Léon Bottou. Large-Scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Physica-Verlag HD, 2010.

[4] Nanette Byrnes. Artificial intolerance, 2016.

[5] Liam Downey. Environmental injustice: Is race or income a better predictor? *Social Science Quarterly*, pages 766–778, 1998.

[6] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.

[7] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[8] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

[11] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2012.

[12] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*, pages 656–666, 2017.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[14] Lauren Kirchner. When discrimination is baked into algorithms. *The Atlantic*, (September 6), 2015.

[15] Min Kyung Lee. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1): 2053951718756684, 2018.

[16] Claire Cain Miller. Can an algorithm hire better than a human. *The New York Times*, 25, 2015.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[18] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.

[19] Cynthia Rudin. Predictive policing using machine learning to detect patterns of crime. *Wired Magazine, August*, 2013.