이학박사학위논문

Two Issues in Classification: Fairness and Extremely Imbalanced Classifications

분류문제에서의 두 가지 이슈: 공정성 및 극단적 불균형 분류문제

2021년 8월

서울대학교 대학원

통계학과

김 사 라

Two Issues in Classification:
Fairness and Extremely Imbalanced Classifications

분류문제에서의 두 가지 이슈:
공정성 및 극단적 불균형 분류문제

지도교수 김 용 대

이 논문을 이학박사 학위논문으로 제출함
2021년 4월

서울대학교 대학원
통계학과
김 사 라

김사라의 이학박사 학위논문을 인준함
2021년 6월

위 원 장	장 원 철
부위원장	김 용 대
위 원	PARK JUN YONG
위 원	원 중 호
위 원	유 규 상

Two Issues in Classification: Fairness and Extremely Imbalanced Classifications

By

Sarah Kim

A Thesis
Submitted in fulfillment of the requirement
for the degree of
Doctor of Philosophy
in Statistics

Department of Statistics
College of Natural Sciences
Seoul National University
August, 2021

ABSTRACT

Two Issues in Classification: Fairness and Extremely Imbalanced Classifications

Sarah Kim
The Department of Statistics
The Graduate School
Seoul National University

In this thesis, we deal with two issues that arise when solving classification problems. The first of which is fairness in artificial intelligence (AI). As they have a vital effect on social decision-making, AI algorithms not only should be accurate and but also should not pose unfairness against certain sensitive groups (e.g., non-white, women). Various specially designed AI algorithms to ensure trained AI models to be fair between sensitive groups have been developed. On the other hand, individual fairness emerged because an AI model that is between-group fair can treat individuals unfairly. However, to find individual-fair algorithms in practice, one must first specify metrics between individuals. Unfortunately, this can be vague and hard to understand in most tasks. In this thesis, we introduce a better guide to between-group fairness, so-called *within-group fairness*, which requires that AI models be fair for those in a same sensitive group and those in different sensitive groups. Within-group fairness leads to training an AI model that satisfies between-group fairness and individual fairness in the same sensitive group. We materialize the concept of within-group fairness by proposing corresponding mathematical definitions and developing learning algorithms to control within-group fairness and between-group fairness simultaneously. Numerical studies show that the proposed learning algorithms improve within-group fairness without sacrificing accuracy as well as between-group fairness.

The second is the classification problem when the imbalance between classes is severe. Imbalanced machine learning problem widely studies in various areas, including fraud detection, medical diagnosis, etc. If there is an imbalance between the classes in the data set, the machine learning algorithm learns with more weights for classes with many examples and fewer weights for classes with few examples. Intuitive and easy-to-use sampling methods such as random oversampling (ROS) have been studied to resolve the imbalance problem. However, simple ROS does not help learn a classifier with better performance, especially in extremely imbalanced problems. In this thesis, we propose a new data augmentation procedure *MixupROS* motivated by Mixup and classification algorithms based on a supervised anomaly detection method. MixupROS uses information from a major class to generate virtual examples belonging to a minor class. Meanwhile, data-level methodologies have limitations in improving classifier performance when it is extremely imbalanced. Hence, we develop algorithms that are extensions of the DeepSAD algorithm for extremely imbalanced problems. Numerical studies on various imbalanced benchmark datasets and CIFAR-10 show that our proposed methods outperform existing methods.

Keywords: Classification, Fairness in artificial intelligence, Within group fairness, Imbalanced classification, Data augmentation, Supervised anomaly detection

Student Number: 2014-22358

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning is a powerful learning method that performs well in various applications such as image classification, machine translation, credit score predictions, etc. Machine learning algorithms have been evolved rapidly over the past decades and offer the best performance for most applications. However, despite its excellent performance, there are some challenges with using machine learning models. This thesis points out two problems when working with machine learning classification models: (i) fairness artificial intelligence; (ii) extremely unbalanced classification.

Fairness artificial intelligence (FAI) is an area where research has recently begun, and several studies have suggested that AI may impose unfairness on some demographic groups [Kleinberg et al., 2018; Mehrabi et al., 2019]. For example, in the recidivism assessment task, AI predicted that blacks would have a higher recidivism rate than whites. Since AI is increasingly being applied to social decision-making, AI algorithms should be accurate and not pose unfairness against certain sensitive groups (e.g., non-whites, women). However, since AI has been trained on datasets with historical biases, trained AI models tend to bias or injustice.

Various learning algorithms to find a fair classifier have been proposed. Generally speaking, they are trying to search for a classifier that is accurate and similar between sensitive groups. For an example of similarity, Hardt et al. [2016] suggested that a classifier's true positive rates for each group are similar. Unfortunately, a fair classifier similar between sensitive groups could be unfair in a specific sensitive group. For this reason, individual fairness [Dwork et al., 2012] has been proposed. However, to find individual fair algorithms in practice, one must first specify metrics between individuals, which is a very problematical task.

Chapter 2 introduces a better guide to group fairness, so-called *within-group fairness*, which requires fairness between and within sensitive groups. Within-group fairness leads to training a classifier that meets group fairness and individual fairness in the same sensitive group. The concept of within-group fairness is materialized by proposing a corresponding mathematical definition and developing a learning algorithm to control both between-group and within-group fairness at the same time. Numerical studies show that the proposed learning algorithm improves within-group fairness without sacrificing accuracy and between-group fairness.

Another problem with using machine learning classification models arises when class imbalances exist. Imbalanced machine learning problem widely studies in various areas, including fraud detection, medical diagnosis, etc. [Phua et al., 2010; Warriach and Tei, 2013]. In general, machine learning models perform poorly when training data is extremely imbalanced between classes. This is because the machine learning model's training process focuses on the major class, a class with many samples. Hence it is difficult for the classifier to learn features of the minor class, which is a class with few samples [Garcia et al., 2007; He and Garcia, 2009; Visa and Ralescu, 2005]. Therefore, it requires additional techniques to learn information from the minor class. To solve the imbalance problem, several algorithms have been proposed [Johnson and Khoshgoftaar, 2019; Krawczyk, 2016].

In Chapter 3, we propose a new data augmentation procedure, *MixupROS*, motivated by Mixup [Zhang et al., 2018] and classification algorithms based on a supervised anomaly detection method [Ruff et al., 2019]. MixupROS uses the information from the major class to create a virtual example belonging to the minor class. On the other hand, data-level methodologies, including oversampling and data augmentation, have limitations in improving classifier performance in extremely imbalanced cases. Hence, we develop algorithms that are extensions of the DeepSAD algorithm [Ruff et al., 2019] for extremely imbalanced problems. Numerical studies on various benchmark datasets verify that our proposed methods outperform existing methods. Concluding remarks follow in Chapter 4.

Chapter 2

Within-group fairness: A guided to better Between-group fairness

2.1 Introduction

Recently, AI (Artificial Intelligence) is being used as decision-making tools in various domains such as credit scoring, criminal risk assessment, education of college admissions [Angwin et al., 2016]. As AI has a wide range of influences on human social life, issues of transparency and ethics of AI are emerging. However, it is widely known that due to the existence of historical bias in data against ethics or regulatory frameworks for fairness, trained AI models based on such biased data could also impose bias or unfairness against a certain sensitive group (e.g., non-white, women)

[Kleinberg et al., 2018; Mehrabi et al., 2019]. Therefore, designing an AI algorithm that is accurate and fair simultaneously has become a crucial research topic.

Demographic disparities due to AI, which refer to socially unacceptable bias that an AI model favors certain groups (e.g., white, men) over other groups (e.g., black, women), have been observed frequently in many applications of AI such as COMPAS recidivism risk assessment [Angwin et al., 2016], Amazon's prime free same-day delivery [Ingold and Soper, 2016],credit score evaluation [Dua and Graff, 2017] to name just a few. Many studies have been done recently to develop AI algorithms that remove or alleviate such demographic disparities in trained AI models to treat sensitive groups as equally as possible. In general, these methods try to search AI models that are accurate and similar between sensitive groups in a certain sense [Zafar et al., 2019]. From now on, criteria of fairness requiring similarity between sensitive groups are referred to as *between-groups fairness* (BGF).

However, BGF may be unfair at the individual level, as it can be satisfied with simple statistical parity (e.g., positive rates) between sensitive groups. For example, if one enforces BGF, an unqualified individual can inadvertently result in positive outcomes, while a more qualified individual can result in negative outcomes. These concerns bring about a concept of *individual fairness* [Dwork et al., 2012]: 'similar' people are treated similarly in classification outcomes. Various studies [Dwork et al., 2012; Lahoti et al., 2019; Yona and Rothblum, 2018] have proposed individual-level fairness metrics under a task-specific metric for the similarity between individuals and their learning algorithms. But, in practice, calculating task-specific metrics between individuals is unclear and difficult to understand, making it socially unacceptable. In this sense, BGF is a more appealing concept for fairness AI, although it ignores individual unfairness.

This thesis considers a better guide to BGF, so-called *within-group fairness* (WGF), which conceptualizes individual unfairness in the same sensitive group when trying to enforce BGF into AI algorithms. Generally speaking, within-group unfairness occurs when an individual is positively treated compared to others in the same sensitive group by an AI model trained without BGF constraints but becomes negatively treated by an AI model trained with BGF constraints.

For an illustrative example of WGF, consider a college admission problem where gender (men vs. women) is a sensitive variable. Let $\mathbf{X}$ and $Y \in \{0, 1\}$ be the input vector and the corresponding output label where $\mathbf{X}$ represents the information of a candidate student such as GPA at high school, SAT score, etc., and Y is the admission result where 0 and 1 mean the rejection and acceptance of the college admission, respectively. The Bayes classifier accepts a student with $\mathbf{X} = \mathbf{x}$ when $\Pr(Y = 1|\mathbf{X} = \mathbf{x}) > 1/2$. Suppose that there are two women 'A' and 'B' with the input vectors $\mathbf{x}_A$ and $\mathbf{x}_B$, respectively and the AI model trained without BGF constraints estimates $\Pr(Y = 1|\mathbf{X} = \mathbf{x}_B) > \Pr(Y = 1|\mathbf{X} = \mathbf{x}_A)$. Then, within-group unfairness occurs when an AI model trained with BGF constraints results in $\Pr(Y = 1|\mathbf{X} = \mathbf{x}_A) > \Pr(Y = 1|\mathbf{X} = \mathbf{x}_B)$. In this situation, which is illustrated in the left panel of Figure 3.1, 'B' could claim that the AI model trained with BGF constraints mistreats her and so it is unfair. We will show in Section 2.5 that there exists non-negligible within-group unfairness in AI models trained on real data with BGF constraints.

Within-group unfairness arises because most existing learning algorithms for BGF force certain statistics (e.g., rate of positive prediction, misclassification error rate, etc.) of a trained AI model being similar across sensitive groups but do not care about what happens to individuals in the same sensitive group at all. For within-group fairness, a desirable AI model is expected at least to preserve the ranks between $\Pr(Y = 1|\mathbf{X} = \mathbf{x}_A)$ and $\Pr(Y = 1|\mathbf{X} = \mathbf{x}_B)$ regardless of estimating $\Pr(Y = 1|\mathbf{X} = \mathbf{x})$ with or without BGF constraints, which is depicted in the right panel of Figure 3.1. Thus, in the same sensitive group, it ensures that more qualified individuals get positive outcomes than less qualified individuals. In this sense, within-group fairness helps to find AI models that are not only between-group fair but also individually fair in the same sensitive group.

Our contributions are three folds. We first define the concept of WGF rigorously. Then we develop learning algorithms that compromise BGF and WGF, as well as accuracy. Finally, we show empirically that the proposed learning algorithms improve WGF while maintaining accuracy and BGF.

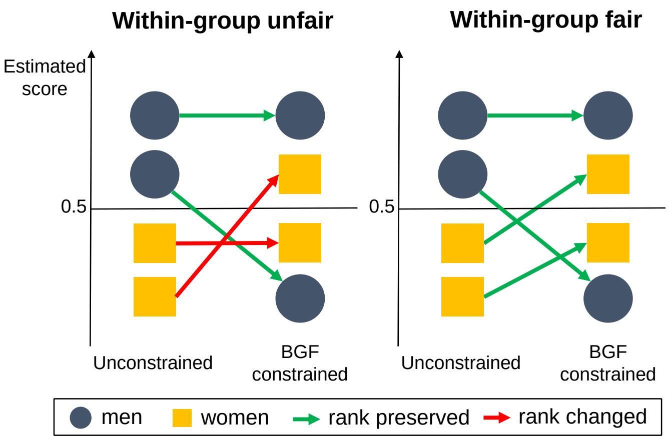

Figure 2.1: A toy example of within-group unfairness: The left panel: without BGF constraints, there exists unfairness against the women sensitive group, but with BGF constraints, the scores of the two women become reversed and thus within-group unfairness occurs. The right panel: the scores of the two women increase together to achieve BGF without within-group unfairness.

Remark. One may argue that training data are prone to bias due to historical prejudices and discriminations, and hence a trained AI model is also biased and socially unacceptable. On the other hand, a trained AI model with BGF constraints does not have such bias and is socially acceptable. Therefore, it would be by no means reasonable to claim unfairness based on discrepancies between socially unacceptable and acceptable AI models. However, note that historical bias in training data is about bias between sensitive groups but not for individuals in the same sensitive group.

For WGF, we implicitly assume that no historical bias among individuals in the same sensitive group exists in training data, which is not too absurd, and thus there is no reason for a trained AI model without BGF constraints to treat individuals in the same sensitive group unfairly. This assumption, of course, needs more debates which we leave as future work.

In Section 2.2, we briefly review methods for BGF, and in Sections 2.3 and 2.4, we propose mathematical definitions of WGF and develop corresponding learning algorithms for classifiers and score functions, respectively. The results of numerical studies are presented in Section 2.5, and remarks about reflecting WGF to pre- and post-processing algorithms for BGF are given in Section 2.6.

2.2 Review of between-group fairness

The concept of WGF is a by-product of BGF, and thus it is helpful to review learning methods for BGF. In this section, we review the definitions of BGF and related studies.

We let $\mathcal{D} = \{(\mathbf{x}_i, z_i, y_i)\}_{i=1}^{n}$ be a set of training data of size n which are independent copies of a random vector $(\mathbf{X}, Z, Y)$ defined on $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^p$. We consider a binary classification problem, which means $\mathcal{Y} = \{0, 1\}$, and for notational simplicity, we let $\mathcal{Z} = \{0, 1\}$, where $Z = 0$ refers to the unprivileged group and $Z = 1$ refers to the privileged group. Whenever the probability is mentioned, we mean it by either the probability of $(\mathbf{X}, Z, Y)$ or its empirical counterpart unless there is any confusion.

In this thesis, we consider AI algorithms which yield a real-valued function $f : \mathcal{X} \to \mathbb{R}$ so-called a score function which assigns positive labeled instances higher scores than negative labeled instances. An example of the score function is the conditional class probability $\Pr(Y = 1|\mathbf{x} = \mathbf{x})$. In most human-related decision makings, real-valued score functions are popularly used (e.g., scores for credit assessment).

Let $\mathcal{F}$ be a given set of score functions, in which we search an optimal score function in a certain sense (e.g., minimizing the cross-entropy for classification problems). Examples of $\mathcal{F}$ are linear functions, reproducing kernel Hilbert space and deep neural networks to name a few. For a given $f \in \mathcal{F}$, the corresponding classifier C_f is defined as $C_f(\mathbf{x}) = \mathbb{1}(f(\mathbf{x}) > 0)$.

2.2.1 Definition of between-group fairness

For a given score function f and a sensitive group $Z = z$, we consider the group performance function of f given as

$$q_z(f) := \mathbb{E}(\mathcal{E}|\mathcal{E}', Z = z) \tag{2.1}$$

for events $\mathcal{E}$ and $\mathcal{E}'$ that might depend on $f(\mathbf{X})$ and Y. The group performance function q_z in (2.1), which is considered by Celis et al. [2019], includes various performance functions used in fairness AI. We summarize representative group performance functions having the form of (2.1) in Table 2.1.

For given group performance functions $q_z(\cdot), z \in \{0, 1\}$, we say

Table 2.1: Some group performance functions

Fairness criteria	$\mathcal{E}$	$\mathcal{E}'$
Disparate impact [Barocas and Selbst, 2016]	$\mathbb{1}\{C_f(X) = 1\}$	$\emptyset$
Equal opportunity [Hardt et al., 2016]	$\mathbb{1}\{C_f(X) = 1\}$	$\{Y = 1\}$
Disparate mistreatment w.r.t. Error rate [Zafar et al., 2019]	$\mathbb{1}\{C_f(X) \neq Y\}$	$\emptyset$
Mean score parity [Coston et al., 2019]	$f(X)$	$\emptyset$

that f satisfies the BGF constraint with respect to q_z if $q_0(f) = q_1(f)$. A relaxed version of the BGF constraint, so-called the ϵ-BGF constraint, is frequently considered, which requires $|q_0(f) - q_1(f)| < \epsilon$ for a given $\epsilon > 0$. Typically, AI algorithms search an optimal function f among those satisfying the ϵ-BGF constraint with respect to given group performance functions $q_z(\cdot), z \in \{0, 1\}$.

2.2.2 Related works

Several learning algorithms have been proposed to find an accurate model f satisfying a given BGF constraint, which are categorized into three groups. In this subsection, we review some methods for each group.

Pre-processing methods: Pre-processing methods remove bias in training data or find a fair representation with respect to sensitive variables before the training phase and learn AI models based on de-biased data or fair representation [Calmon et al., 2017;

Dixon et al., 2018; Feldman et al., 2015; Kamiran and Calders, 2012; Webster et al., 2018; Xu et al., 2018; Zemel et al., 2013]. Kamiran and Calders [2012] suggested pre-processing methods to eliminate bias in training data by use of label changing, reweighing and sampling. Based on the idea that transformed data should not be able to predict the sensitive variable, Feldman et al. [2015] proposed a transformation of input variables for eliminating the disparate impact. To find a fair representation, Calmon et al. [2017]; Zemel et al. [2013] proposed a data transformation mapping for preserving accuracy and alleviating discrimination simultaneously. Pre-processing methods for fair learning on text data were studied by Dixon et al. [2018]; Webster et al. [2018].

In-processing methods: In-processing methods generally train an AI model by minimizing a given cost function (e.g., the cross-entropy, the sum of squared residuals, the empirical AUC etc.) subject to a ϵ-BGF constraint. Most group performance functions $q_z(\cdot)$ are not differentiable, and thus various surrogated group performance functions and corresponding ϵ-BGF constraints have been proposed [Celis et al., 2019; Cho et al., 2020; Donini et al., 2018; Goh et al., 2016; Kamishima et al., 2012; Menon and Williamson, 2018; Narasimhan, 2018; Vogel et al., 2020; Zafar et al., 2017, 2019]. Kamishima et al. [2012] used a fairness regularizer which is an approximation of the mutual information between the sensitive variable and the target variable. Zafar et al. [2017, 2019] proposed covariance-type fairness constraints as tractable proxies targeting the disparate impact and the equality of the false positive or negative rate, and Donini et al. [2018] used a linear surrogated group performance function for the equalized odds. On the other hand, Celis et al. [2019]; Menon and Williamson [2018] derived an optimal classifier for a constrained fair classification as a form of an instance-dependent threshold. Also, for fair score functions, Vogel et al. [2020] proposed fairness constraints based on ROC curves of each sensitive group.

Post-processing methods: Post-processing methods first learn an AI model without any BGF constraint and then transform the decision boundary or score function of the trained AI model for each sensitive group to satisfy given BGF criteria [Corbett-Davies et al., 2017; Fish et al., 2016; Hardt et al., 2016; Jiang et al., 2020; Kamiran et al., 2012; Pleiss et al., 2017; Wei et al., 2020]. Chzhen et al. [2019]; Hardt et al. [2016] suggested finding sensitive group dependent thresholds to get a fair classifier with respect to equal opportunity. Jiang et al. [2020]; Wei et al. [2020] developed an algorithm to transform the original score function to achieve a BGF constraint.

2.3 Within-group fairness for classifiers

We assume that there exists a known optimal classifier $C^\star$ which could be the Bayes classifier or its estimate. For example, we can use $C_{f^\star}$ for $C^\star$, where $f^\star$ is the unconstrained minimizer of the cross-entropy on $\mathcal{F}$.

We mostly focus on in-processing methods for the BGF and explain how to reflect WGF into a learning procedure. Remarks about how to reflect WGF to pre- and post-processing methods are given in Section 2.6.

2.3.1 Definition of within-group fairness

Conceptually, WGF means that the classifier C_f and $C^\star$ have the same ranks in each sensitive group. That is, for two individuals $\mathbf{x}_A$ and $\mathbf{x}_B$ in the same sensitive group with $C^\star(\mathbf{x}_A) > C^\star(\mathbf{x}_B)$, WGF requires that $C_f(\mathbf{x}_A) \geq C_f(\mathbf{x}_B)$. To materialize this concept of WGF, we define the WGF constraint as

$$\begin{aligned} &\Pr\left\{C^\star(\mathbf{X}) = 0, C_f(\mathbf{X}) = 1 | Z = z\right\} = 0 \\ \text{or } &\Pr\left\{C^\star(\mathbf{X}) = 1, C_f(\mathbf{X}) = 0 | Z = z\right\} = 0 \end{aligned} \tag{2.2}$$

for each $z \in \{0, 1\}$. Similar to the BGF, we relax the constraint (2.2) by requiring that either of the two probabilities is small. That is, we say that f satisfies the δ-WGF constraint for a given $\delta > 0$ if

$$\max_{z \in \{0,1\}} \min\{a_{01|z}(f), a_{10|z}(f)\} < \delta, \tag{2.3}$$

where $a_{ij|z}(f) = \Pr\{C^\star(X) = i, C_f(X) = j | Z = z\}$.

2.3.2 Directional within-group fairness

Many BGF constraints have their own implicit directions toward which the classifier is expected to be guided in the training phase. We can design a special WGF constraint reflecting the implicit direction of a given BGF constraint, resulting in more desirable classifiers (better guided, fairer, and frequently more accurate). Below, we present two such WGF constraints.

Disparate impact: Note that the disparate impact requires that

$$\Pr\{C_f(\mathbf{X}) = 1|Z = 0\} = \Pr\{C_f(\mathbf{X}) = 1|Z = 1\}.$$

Suppose that $\Pr\{C^\star(\mathbf{X}) = 1|Z = 0\} < \Pr\{C^\star(\mathbf{X}) = 1|Z = 1\}$. Then, we expect that a desirable classifier C_f achieves this BGF constraint by increasing $\Pr\{C_f(\mathbf{X}) = 1|Z = 0\}$ from $\Pr\{C^\star(\mathbf{X}) = 1|Z = 0\}$ and decreasing $\Pr\{C_f(\mathbf{X}) = 1|Z = 1\}$ from $\Pr\{C^\star(\mathbf{X}) = 1|Z = 1\}$. To reflect this direction, we can enforce a learning algorithm to search a classifier C_f satisfying $\Pr\{C^\star(\mathbf{X}) = 1|Z = 0\} < \Pr\{C_f(\mathbf{X}) = 1|Z = 0\}$ and $\Pr\{C^\star(\mathbf{X}) = 1|Z = 1\} > \Pr\{C_f(\mathbf{X}) = 1|Z = 1\}$. Based on this argument, we define the directional δ-WGF constraint for the disparate impact as

$$\max\{a_{10|0}(f), a_{01|1}(f)\} < \delta. \tag{2.4}$$

Equal opportunity: The equal opportunity constraint is given as

$$\Pr\{C_f(\mathbf{X}) = 1|Z = 0, Y = 1\} = \Pr\{C_f(\mathbf{X}) = 1|Z = 1, Y = 1\}.$$

Suppose that $\Pr\{C^\star(\mathbf{X}) = 1|Z = 0, Y = 1\} < \Pr\{C^\star(\mathbf{X}) = 1|Z = 1, Y = 1\}$. A similar argument for the disparate impact leads us to define the directional δ-WGF constraint for the equal opportunity as

$$\max\{a_{10|01}(f), a_{01|11}(f)\} < \delta \tag{2.5}$$

and

$$\max_{z\in\{0,1\}} \min\left\{a_{10|z0}(f), a_{01|z0}(f)\right\} < \delta, \tag{2.6}$$

where

$$a_{ij|zy}(f) = \Pr\{C^\star(X) = i, C_f(X) = j|Z = z, Y = y\}.$$

2.3.3 Learning with doubly-group fairness constraints

We say that f satisfies the (ϵ, δ)-doubly-group fairness constraint if $B(f) < \epsilon$ and $W(f) < \delta$, where B is a given BGF constraint and W is the corresponding WGF constraint proposed in the previous two subsections. In this section, we propose a relaxed version of $W(\cdot)$ for easy computation. As we review in Section 2.2, many relaxed versions of $B(\cdot)$ have been proposed already.

The WGF constraints considered in Sections 2.3.1 and 2.3.2 are hard to be used as themselves in the training phase since they are neither convex nor continuous. A standard approach to resolve this problem is to use a convex surrogated function. For example, a surrogated version of the WGF constraint (2.3) is $W_{\text{surr}}(f) < \delta$, where

$$\begin{aligned} W_{\text{surr}}(f) := \max_{z\in\{0,1\}} \min\Big\{&\mathbb{E}\left\{\phi(-f(\mathbf{X}))|Z = z, Y^\star = 1\right\} p_{1|z}, \\ &\mathbb{E}\left\{\phi(f(\mathbf{X}))|Z = z, Y^\star = 0\right\} p_{0|z}\Big\}, \end{aligned} \tag{2.7}$$

where $Y^\star = C^\star(\mathbf{X}), p_{y|z} = \Pr(C^\star(\mathbf{X}) = y|Z = z)$ and ϕ is a convex surrogated function of the indicator function $\mathbb{1}(z \geq 0)$. In this thesis, we use the hinge function given as $\phi_{\text{hinge}}(z) = (1 + z)_+$ as a convex surrogated function which is popularly used for fair AI [Donini et al., 2018; Goh et al., 2016; Wu et al., 2018]. The surrogated versions for the other WGF constraints are derived similarly. Finally, we estimate f by $\hat{f}$ that minimizes the regularized cost function

$$\mathcal{L}(f) + \lambda B_{\text{surr}}(f) + \eta W_{\text{surr}}(f), \tag{2.8}$$

where $\mathcal{L}$ is a given cost function (e.g., the cross-entropy) and B_{surr} and W_{surr} are the surrogated constraints of B and W, respectively. The nonnegative constants λ and η are regularization parameters which are selected so that $\hat{f}$ satisfies $B(\hat{f}) < \epsilon$ and $W(\hat{f}) < \delta$.

2.3.4 Related notions with within-group fairness

There are several fairness concepts which are somehow related to WGF. However, the existing concepts are quite different from our WGF.

1. Unified fairness: Speicher et al. [2018] used the term 'within-group fairness'. However, WGF of Speicher et al. [2018] is different from our WGF. Speicher et al. [2018] measured individual-level benefits of a given prediction model and they defined the model to be WGF if the individual benefits in each group are similar. They also illustrated that WGF keeps decreasing as BGF increases. Our WGF is nothing to do with individual-level benefits. Our WGF can be high even when individual-level benefits are not similar. Also, our WGF can

increase even when BGF increases.

2. Slack consistency: Nachum and Jiang [2019] proposed the 'slack consistency' which requires that the estimated scores of each individual should be monotonic with respect to slack variables used in fairness constraints. Slack consistency does not guarantee within-group fairness because the ranks of the estimated scores can change even when they move monotonically.

2.4 Within-group fairness for score functions

Similarly to classifiers, the WGF for score functions requires that $f(\mathbf{x}_A) > f(\mathbf{x}_B)$ when $f^\star(\mathbf{x}_A) > f^\star(\mathbf{x}_B)$ and vice versa for two individuals $\mathbf{x}_A$ and $\mathbf{x}_B$ in the same sensitive group, where $f^\star$ is a known optimal score function such as the conditional class probability $\Pr(Y = 1|\mathbf{X})$ or its estimate. To realize this concept, we define the WGF constraint for a score function f as $\tau_z(f) = 1$ for $z \in \{0, 1\}$, where $\tau_z(\cdot)$ is the Kendall's τ between f and $f^\star$ conditional on $Z = z$, that is

$$\tau_z(f) = \mathbb{E}_{(\mathbf{X}_1,\mathbf{X}_2)}\Big[\mathbb{1}\{(f(\mathbf{X}_1)-f(\mathbf{X}_2))(f^\star(\mathbf{X}_1)-f^\star(\mathbf{X}_2)) > 0\}\Big| Z = z\Big],$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are independent copies of $\mathbf{X}$. In turn, the δ-WGF constraint for a score function f is $1 - \tau_z(f) < \delta, z \in \{0, 1\}$.

Similar to classifiers, we need a convex surrogated version of the δ-WGF constraint, and a candidate would be $1 - \tau_{\phi,z}(f) <$

$\delta, z \in \{0, 1\}$, where

$$\tau_{\phi,z}(f) = 1 - \mathbb{E}_{(\mathbf{X}_1,\mathbf{X}_2)}\left[\phi\{(f(\mathbf{X}_1) - f(\mathbf{X}_2))(f^\star(\mathbf{X}_1) - f^\star(\mathbf{X}_2))\}\Big| Z = z\right]$$

and ϕ is a convex surrogated function of $\mathbb{1}(z > 0)$ such as the ϕ_{hinge}.

However, there are problems in using $\tau_{\phi,z}(f)$ for learning a WGF score function. The $f^\star$, which is perfectly fair (i.e., $\tau_z(f^\star) = 1$), may not be perfectly fair in terms of $\tau_{\phi,z}(f)$ in the sense that there exists $f \in \mathcal{F}$ such that $f(\cdot) \not\equiv f^\star(\cdot)$ and $\tau_{\phi,z}(f) > \tau_{\phi,z}(f^\star)$. In fact, if $\|f^\star\|_\infty \leq 1/2$, then and $f = \alpha f^\star$ for any $\alpha > 1$ has a larger value of $\tau_{\phi,z}$ than $f^\star$. In addition, a surrogated penalty of WGF is computed over all pairs of the training data, hence it requires huge computations.

To resolve these problems, we propose an alternative convex surrogated version of δ-WGF. Let $\mathcal{M}$ be the set of all monotonically increasing functions from $\mathbb{R} \to \mathbb{R}$. Then any $f = m \circ f^\star$, $m \in \mathcal{M}$ satisfies $\tau_z(f) = 1$, $z \in \{0, 1\}$. Let $\mathcal{F}_w = \{f \in \mathcal{F} : \tau_z(f) = 1, z \in \{0, 1\}\}$. For a convex surrogated δ-WGF constraint, we construct a sequence of subsets $\{\mathcal{F}_{w,\delta}, \delta \geq 0\}$ of $\mathcal{F}$ satisfying: (i) $\mathcal{F}_{w,\delta}$ is increasing (i.e., $\mathcal{F}_{w,\delta_1} \subset \mathcal{F}_{w,\delta_2}$) for $\delta_1 \leq \delta_2$, (ii) $\mathcal{F}_{w,0} = \mathcal{F}_w$ and (iii) $\mathcal{F}_{w,\delta}$ is a convex set.

Let $J : \mathcal{F} \to [0, \infty)$ be a measure of complexity of functions in $\mathcal{F}$ such that J is convex and $f(\cdot) \equiv 0$ (almost everywhere) if $J(f) = 0$. Two examples of J are $J(f) = \|\beta\|_2^2$ if $\mathcal{F} = \{f(\mathbf{x}) = \mathbf{x}^\top\beta, \beta \in \mathbb{R}^p\}$ and $J(f) = \|f\|^2_{\mathcal{H}_K}$ when $\mathcal{F} = \mathcal{H}_K$, the reproducing kernel Hilbert space generated by a kernel K. For a given complexity measure J, we consider the set

$$\mathcal{F}_{w,\delta} = \{m \circ f^{\star}(\cdot) + g(\cdot) : m \in \mathcal{M}, g \in \mathcal{F}, J(g) \leq \delta\}.$$

We say that f satisfies the (convex surrogated) δ-WGF constraint if $f \in \mathcal{F}_{w,\delta}$.

To learn a (ϵ, δ)-doubly fair score function, we minimize

$$L(m \circ f^{\star} + g) + \lambda B(m \circ f^{\star} + g) + \eta J(g) \tag{2.9}$$

with respect to $m \in \mathcal{M}$ and $g \in \mathcal{F}$, where $B(\cdot)$ is a given BGF constraint and λ, η are regularization parameters. In the following two subsections, we explain in detail how to implement the above learning algorithm for specific choices of $\mathcal{F}$. For $f^{\star}$, we use the unconstraint minimizer of $L(f)$ on $\mathcal{F}$.

Linear models

Suppose that $\mathcal{F}$ is the set of linear functions. It can be shown that any $f \in \mathcal{F}$ satisfying $\tau_z(f) = 1$ can be expressed by $f(\cdot) = a + bf^{\star}(\cdot)$ for some $a \in \mathbb{R}$ and $b \in (0, \infty)$. Hence, $\mathcal{M}$ is the set of linear functions with positive trends. For the penalty J, we can use any penalty function used for the linear models such as Lasso and ridge penalties. In this thesis, we use the ridge penalty since computation is easier. Finally, we estimate the score function by minimizing

$$L(a + bf^{\star} + g_{\beta}) + \lambda_1 B_{conv}(a + bf^{\star} + g_{\beta}) + \eta \|\beta\|_2^2$$

with respect to $a \in \mathbb{R}, b \in (0, \infty)$ and $\beta \in \mathbb{R}^p$, where $g_{\beta}(\mathbf{x}) = \mathbf{x}^{\top}\beta$.

Nonparametric regression

Let $\mathcal{F}$ be a set of all measurable functions on $\mathbb{R}^p$ satisfying a certain smoothness condition. For example, suppose that $\mathcal{F}$ is the set of Hölder smooth functions of order r. Then, $\mathcal{M}$ includes monotonically increasing functions satisfying the Hölder smoothness of order $s \geq r$.

We propose to estimate m and g based on the modified gradient descent algorithm given as follows. Let

$$C(m, g) = L_n(m \circ f^\star + g) + \lambda B(m \circ f^\star + g) + \eta J(g).$$

We minimize $C(m, g)$ with respect to m while g is fixed and then minimize $C(m, g)$ with respect to g with m being fixed. We first explain how to estimate m while g is fixed. Let m^{curr} be the current estimate of m. For given i, let

$$\nabla_i^{\text{curr}} = \left.\frac{\partial C}{\partial m \circ f^\star(\mathbf{x}_i)}\right|_{m \circ f^\star(\mathbf{x}_i) = m^{\text{curr}} \circ f^\star(\mathbf{x}_i)}.$$

Then, we update m^{curr} by $\hat{m}$ which minimizes

$$\sum_{i=1}^{n} \{\hat{m}_i - m \circ f^\star(\mathbf{x}_i)\}^2$$

with respect to $m \in \mathcal{M}$, where $\hat{m}_i = m^{\text{curr}} \circ f^\star(x_i) - \gamma \nabla_i^{\text{curr}}$ for a given step size $\gamma > 0$. If $\mathcal{M}$ consists of monotonically increasing functions having the first derivative, the estimation $\hat{m}$ obtained by PAVA (Pool Adjacent Violator algorithm)[Barlow, 1972; Mair et al., 2009] can be used. For estimation of the monotonically increasing function with smoothness order $r > 1$, there are several works based on splines with monotonically increasing constraints

[Mammen and Thomas-Agnan, 1999; Pya and Wood, 2015; Wang and Li, 2008].

Estimation of g with a fixed m can be done by use of appropriate nonparametric regression techniques such as the generalized additive model [Hastie and Tibshirani, 1990], boosting [Freund, 1995; Friedman, 2001] and deep learning [Goodfellow et al., 2016; LeCun et al., 2015; Schmidt-Hieber et al., 2020].

2.5 Numerical studies

We investigate the impacts of the WGF constraints on the prediction accuracy as well as the BGF by analyzing real-world datasets. We consider linear logistic and deep neural network (DNN) models for $\mathcal{F}$ and use the cross-entropy for $\mathcal{L}$. For DNN, fully connected neural networks with one hidden layer and p many hidden nodes are used. We train the models by the gradient descent algorithm [Bottou, 2010] implemented by Python with related libraries `pytorch, scikit-learn, numpy`. The SGD optimizer is used with momentum 0.9 and a learning rate of either 0.1 or 0.01 depending on the dataset. We use the unconstrained minimizer of $\mathcal{L}$ for $f^{\star}$.

Datasets. We analyze four real world datasets, which are popularly used in fairness AI research and publicly available: (i) The Adult Income dataset (*Adult*, Dua and Graff [2017]); (ii) The Bank Marketing dataset (*Bank*, Dua and Graff [2017]); (iii) The Law School dataset (*LSAC*, Wightman and Ramsey [1998]); (iv) The

Compas Propublica Risk Assessment dataset (*COMPAS*, Larson et al. [2016]). Except for the dataset *Adult*, we split the training and test datasets randomly by 8:2 ratio and repeat 5 times training/test splits for performance evaluation.

2.5.1 Within-group fair classifiers

We consider following group performance functions for the BGF: the disparate impact (DI) [Barocas and Selbst, 2016] and the disparate mistreatment w.r.t. error rate [Zafar et al., 2019], which are defined as

$$\begin{aligned} \mathrm{DI}(f) &= |\Pr(C_f(\mathbf{X}) = 1|Z = 1) - \Pr(C_f(\mathbf{X}) = 1|Z = 0)| \\ \mathrm{ME}(f) &= |\Pr(C_f(\mathbf{X}) \neq Y|Z = 0) - \Pr(C_f(\mathbf{X}) \neq Y|Z = 1)|. \end{aligned}$$

Note that the DI is directional while the ME is not. For the surrogated BGF constraints, we replace the indicator function with the hinge function in calculating the BGF constraints as is done by Goh et al. [2016]; Wu et al. [2018]. We name the corresponding BGF constraints by Hinge-DI and Hinge-ME respectively. The results for other surrogated constraints such as the covariance type constraints proposed by Zafar et al. [2017, 2019] and the linear surrogated functions considered in Padala and Gujar [2020] are presented in the Appendix. In addition, the results for the equal opportunity constraint are summarized in the Appendix.

For investigating the impacts of WGF on trained classifiers, we first fix the ϵ for each BGF constraint, and we choose the regularization parameters λ and η to make the classifier $\hat{f}$ minimizing the regularized cost function (2.8) satisfy the ϵ-BGF constraint. Then, we assess the prediction accuracy and the degree of WGF of $\hat{f}$.

2.5.1.1 Targeting for disparate impact

Table 2.2 presents the three 2×2 tables comparing the results of the unconstrained DNN classifier ($\hat{Y}^{\star}$) and three DNN classifiers ($\hat{Y}$) trained on the dataset *Adult*: (i) only with the DI constraint, (ii) with the DI and WGF constraints and (iii) with the DI and directional WGF (dWGF) constraints. We let ϵ be around 0.03. The numbers marked in red are subjects treated unfairly with respect to the dWGF. Note that the numbers of unfairly treated subjects are reduced much with the WGF and dWGF constraints and the dWGF constraint is more effective. We report that the accuracies of the three classifiers on the test data are 0.837, 0.840 and 0.839, respectively, which indicates that the WGF and dWGF constraints improve the WGF without hampering the accuracy. Compared to the dWGF, the WGF constraint is less effective, which is observed consistently for different datasets when a BGF constraint is directional. See Table A.2 in the Appendix for the corresponding numerical results. Thus, hereafter we consider the dWGF only for the DI which has an implicit direction.

Table 2.3 summarizes the performances of the three classifiers - $C^{\star}$ and the two classifiers trained with the DI constraint and the DI and dWGF constraints (doubly-fair, DF), respectively. In Table 2.3, we report the accuracies as well as the values of DI and dWGF terms (i.e., $\mathrm{DI}(\hat{f})$ and $\max\{a_{10|0}(\hat{f}), a_{01|1}(\hat{f})\}$, respectively). We observe that the DF classifier improves the dWGF while keeping that the DI values and accuracies are favorably comparable to those of the BGF classifier. For reference, the performances with the WGF constraint are summarized in the Supplementary material.

To investigate the sensitivity of the accuracy to the degree of WGF, the scatter plots between various dWGF values and the corresponding accuracies for the DF linear logistic model are given in Figure 2.2, where the DI value is fixed around 0.03. The accuracies are not sensitive to the dWGF values. Moreover, for the datasets *Adult*, *Bank* and *LSAC*, the accuracies keep increasing as the dWGF value decreases.

While we analyzed the datasets *Bank* and *LSAC*, we found an undesirable aspect of the learning algorithm only with the DI constraint. The corresponding classifiers improve the DI by decreasing (or increasing) the probabilities $P(\hat{Y} = 1|Z = 0)$ and $P(\hat{Y} = 1|Z = 1)$ simultaneously compared to $P(Y^\star = 1|Z = 0)$ and $P(Y^\star = 1|Z = 1)$. A better way to improve the DI would be to increase $P(\hat{Y} = 1|Z = 0)$ and decrease $P(\hat{Y} = 1|Z = 1)$ when $P(Y^\star = 1|Z = 0) < P(Y^\star = 1|Z = 1)$. Figures 2.3 show that this undesirable aspect disappears when the dWGF constraint is considered.

2.5.1.2 Targeting for disparate mistreatment

The results of the performances of the DF classifier with the ME as a BGF constraint are presented in Table 2.4. Since the ME has no implicit direction, we use the undirectional WGF constraint. The overall conclusions are similar to those for the DI and dWGF constraints. That is, the undirectional WGF constraint also works well.

2.5.2 Within-group fair for score function

In this section, we examine the WGF constraint for score functions. We choose AUC (area under the ROC) as evaluation metrics for prediction accuracy. For the BGF, we consider the mean score parity (MSP, Coston et al. [2019]):

$$\mathrm{MSP}(f) = \left|\mathbb{E}(\sigma(f(X))|Z=1) - \mathbb{E}(\sigma(f(X))|Z=0)\right|,$$

where $\sigma : x \mapsto 1/(1+e^{-x})$ is the sigmoid function. To check how much the estimated score function $\hat{f}$ is within-group fair, we calculate Kendall's τ between $\hat{f}$ and the ground-truth score function $f^{\star}$ on the test data for each sensitive group, and then we average them, which is denoted by $\bar{\tau}$ in Table 2.5. We choose the regularization parameters λ and η such that $\bar{\tau}$ of $\hat{f}$ is as close to 1 as possible while maintaining the MSP value around 0.03.

Table 2.5 amply shows that the DF score function always improves the degree of WGF (measured by $\bar{\tau}$) and the accuracy in terms of AUC simultaneously while keeping the degree of BGF at a reasonable level. Here, we consider the DF score function with the surrogated penalty for Kendall's τ, denoted by DF-Surr, and the DF score function in (2.9), denoted by DF-Mono.

With respect to the BCE, the BGF and DF score functions are similar. The superiority of the DF score function in terms of AUC compared with the BGF score function is partly because the WGF constraint shrinks the estimated score toward the ground-truth score (Uncons. in Table 2.5) which is expected to be most accurate. Based on these results, we conclude that the WGF constraint is a useful guide to find a better score function with respect to AUC as well as the WGF.

2.6 Remarks on within-group fairness for pre- and post-processing methods

Various pre- and post-processing methods for fair AI have been proposed. An advantage of these methods compared to constrained methods is that the methods are simple, computationally efficient but yet reasonably accurate. In this section, we briefly explain how to reflect the WGF to pre- and post-processing methods for the BGF.

2.6.1 Pre-processing methods and within-group fairness

Basically, pre-processing methods transform the training data in a certain way to be between-group fair and train an AI model on the transformed data. To reflect the WGF, it suffices to add a WGF constraint in the training phase. Let $\mathcal{D}_{\text{trans}}$ be the transformed training data to be between-group fair and let $\mathcal{L}_{\text{trans}}$ be the corresponding cost function. Then, we learn a model by minimizing $L_{\text{trans}}(f) + \eta W_{\text{conv}}(f)$ for $\eta > 0$.

Table 2.6 presents the results of the models trained on the preprocessing training data and a WGF constraint for various values of η, where the DI is used as the BGF and thus the corresponding dWGF constraint is used. In this experiment, we use the linear logistic model and the Massaging [Kamiran and Calders, 2012] for the pre-processing. Surprisingly we observed that introducing the dWGF constraint to the pre-processing method helps to improve the BGF and WGF simultaneously without sacrificing the accuracies much.

2.6.2 Post-processing methods and within-group fairness

For the BGF score functions, Jiang et al. [2020] developed an algorithm to obtain two monotonically nondecreasing transformations $m_z, z \in \{0, 1\}$ such that $m_0 \circ f^\star$ and $m_1 \circ f^\star$ are BGF in the sense that the distributions of $m_0 \circ f^\star(\mathbf{X})|Z = 0$ and $m_1 \circ f^\star(\mathbf{X})|Z = 1$ are the same. It is easy to check that the transformed score function $m_z \circ f^\star(\mathbf{x})$ is a perfectly WGF score function even though it depends on the sensitivity group variable z. Note that the algorithm in Section 2.4 yields score functions not depending on z.

Table 2.2: Comparison of the results of the three DNN classifiers trained (i) only with the BGF constraint, (ii) with the BGF and WGF constraints and (iii) with the BGF and dWGF constraints on the dataset *Adult*. Marked in red represent the numbers of subjects treated unfairly in a same sensitive group.

Only with the DI constraint					
	$Z = 0$			$Z = 1$	
	$\hat{Y} = 0$	$\hat{Y} = 1$		$\hat{Y} = 0$	$\hat{Y} = 1$
$\hat{Y}^{\star} = 0$	4,592	350	$\hat{Y}^{\star} = 0$	7,966	86
$\hat{Y}^{\star} = 1$	13	466	$\hat{Y}^{\star} = 1$	945	1,863
With the DI and WGF constraints					
	$Z = 0$			$Z = 1$	
	$\hat{Y} = 0$	$\hat{Y} = 1$		$\hat{Y} = 0$	$\hat{Y} = 1$
$\hat{Y}^{\star} = 0$	4,703	239	$\hat{Y}^{\star} = 0$	8,021	31
$\hat{Y}^{\star} = 1$	27	452	$\hat{Y}^{\star} = 1$	1,156	1,652
With the DI and dWGF constraints					
	$Z = 0$			$Z = 1$	
	$\hat{Y} = 0$	$\hat{Y} = 1$		$\hat{Y} = 0$	$\hat{Y} = 1$
$\hat{Y}^{\star} = 0$	4,718	224	$\hat{Y}^{\star} = 0$	8,024	28
$\hat{Y}^{\star} = 1$	18	461	$\hat{Y}^{\star} = 1$	1,178	1,630

Table 2.3: Results for the DF classifier with the Hinge-DI constraint. Except for the dataset *Adult*, the average performances are given.

		Linear model			DNN model		
Dataset	Method	ACC	DI	dWGF	ACC	DI	dWGF
Adult	Uncons.	0.852	0.172	0.000	0.853	0.170	0.000
	Hinge-DI	0.833	0.028	0.005	0.837	0.029	0.008
	Hinge-DI-DF	0.836	0.028	0.003	0.839	0.026	0.003
Bank	Uncons.	0.908	0.195	0.000	0.904	0.236	0.000
	Hinge-DI	0.901	0.024	0.018	0.899	0.029	0.033
	Hinge-DI-DF	0.904	0.021	0.007	0.905	0.029	0.032
LSAC	Uncons.	0.823	0.120	0.000	0.856	0.131	0.000
	Hinge-DI	0.809	0.016	0.014	0.816	0.032	0.064
	Hinge-DI-DF	0.813	0.018	0.009	0.809	0.029	0.047
COMPAS	Uncons.	0.757	0.164	0.000	0.757	0.162	0.000
	Hinge-DI	0.641	0.024	0.153	0.639	0.030	0.142
	Hinge-DI-DF	0.618	0.025	0.145	0.654	0.033	0.120

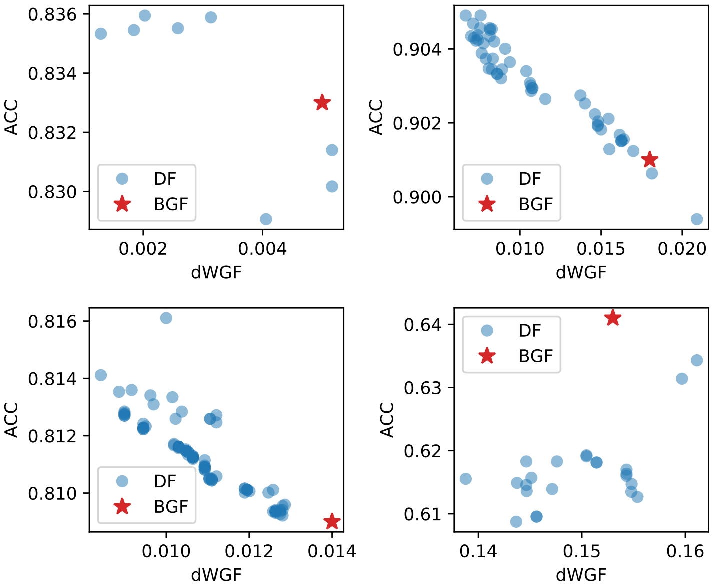

Figure 2.2: Scatter plots of the accuracies and dWGF values for the DF linear regression model with the DI values around 0.03. (Topleft) *Adult*; (Topright) *Bank*; (Bottomleft) *LSAC*; (Bottomright) *COMPAS*. Red star points in each figure represent the results of the BGF classifier.

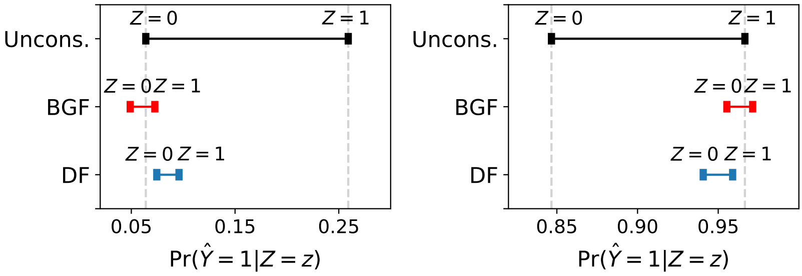

Figure 2.3: Comparison of the conditional probabilities of each group for the datasets *Bank* (Left) and *LSAC*(Right).

Table 2.4: Results for the DF classifier with the Hinge-ME constraint. Except for the dataset *Adult*, average performances are given.

		Linear model			DNN model		
Dataset	Method	ACC	ME	WGF	ACC	ME	WGF
Adult	Uncons.	0.852	0.117	0.000	0.853	0.105	0.000
	Hinge-ME	0.834	0.060	0.005	0.822	0.025	0.059
	Hinge-ME-DF	0.834	0.060	0.005	0.825	0.031	0.026
Bank	Uncons.	0.908	0.177	0.000	0.904	0.174	0.000
	Hinge-ME	0.740	0.044	0.068	0.902	0.164	0.076
	Hinge-ME-DF	0.749	0.045	0.020	0.897	0.165	0.047
LSAC	Uncons.	0.823	0.090	0.000	0.856	0.071	0.000
	Hinge-ME	0.759	0.028	0.038	0.815	0.044	0.040
	Hinge-ME-DF	0.742	0.020	0.017	0.803	0.038	0.001
COMPAS	Uncons.	0.757	0.022	0.000	0.757	0.024	0.000
	Hinge-ME	0.740	0.020	0.018	0.738	0.016	0.018
	Hinge-ME-DF	0.743	0.018	<0.001	0.757	0.017	0.001

Table 2.5: Results of the DF score functions. Except for the dataset *Adult*, averages performances are given.

		Linear model			DNN model		
Dataset	Method	AUC	MSP	$\bar{\tau}$	AUC	MSP	$\bar{\tau}$
Adults	Uncons.	0.905	0.173	1.000	0.907	0.177	1.000
	BGF	0.874	0.027	0.838	0.878	0.032	0.796
	DF-Surr	0.890	0.022	0.944	0.878	0.031	0.847
	DF-Mono	0.876	0.027	0.868	0.888	0.032	0.920
Bank	Uncons.	0.932	0.217	1.000	0.926	0.238	1.000
	BGF	0.899	0.027	0.674	0.922	0.042	0.710
	DF-Surr	0.905	0.031	0.697	0.924	0.032	0.742
	DF-Mono	0.919	0.031	0.758	0.901	0.033	0.818
LSAC	Uncons.	0.732	0.125	1.000	0.831	0.142	1.000
	BGF	0.697	0.022	0.641	0.803	0.026	0.637
	DF-Surr	0.697	0.023	0.642	0.812	0.028	0.693
	DF-Mono	0.712	0.025	0.746	0.771	0.027	0.646
COMPAS	Uncons.	0.822	0.122	1.000	0.825	0.119	1.000
	BGF	0.738	0.026	0.501	0.756	0.030	0.566
	DF-Surr	0.771	0.033	0.617	0.765	0.031	0.598
	DF-Mono	0.779	0.033	0.638	0.748	0.029	0.582

Table 2.6: Comparison of the accuracy and fairnesses of the preprocessing method with and without the dWGF constraint. The results are evaluated on the dataset *Adult*.

Method	η	Acc	DI	dWGF
Massaging	-	0.837	0.069	0.009
Massaging + dWGF	0.5	0.837	0.048	0.004
	1.0	0.836	0.037	0.003

Chapter 3

Methodologies for extremely imbalanced problems

3.1 Introduction

In classification problems, supervised machine learning models need numerous labeled examples for each class to perform well. It is well known that if the numbers of instances in each class are extremely imbalanced, most machine learning methods learned to focus on classes with relatively more examples [Garcia et al., 2007; He and Garcia, 2009; Visa and Ralescu, 2005]. Hence a model learned based on an imbalanced dataset usually has low prediction power in classes with relatively less instance. Hereafter, we denote the ‘major’/‘minor’ class as a class with a large/small number of objects, respectively.

Imbalanced machine learning methods can be used to diagnose diseases like cancer, brain tumor, where the major class is from healthy patients, and the minor class is from having disease patients. Also, it can be used for faulty detecting product, fraud detection, oil-spill detection, etc. [Chan et al., 1999; Kubat et al., 1998; Phua et al., 2010; Warriach and Tei, 2013]. Since, in many applications, instances belonging to a minor class are rare or expensive, a class imbalanced easily occurs, and appropriate analysis is demanded. In general, when a class imbalance exists in a training dataset, an analyst may use the random oversampling method to give sufficient weights to the minor class. Our research shows that a random oversampling method, a simple replication of instances belonging to the minor class, does not work in extremely imbalanced settings. Therefore, it requires other techniques to learn information from the minor class.

Methods handling imbalanced datasets are roughly divided into data-level methods, algorithm-level methods [Johnson and Khoshgoftaar, 2019; Krawczyk, 2016]. Data level methods use a sampling approach to reduce the imbalance between classes, and algorithm-level methods develop a new loss function giving more weights to the minor classes. Several data-level methods used random oversampling (ROS) [Ando and Huang, 2017; Chawla et al., 2002; Hensman and Masko, 2015; Lee et al., 2016; Nickerson et al., 2001] that generates new instances in the minor class based on repetition, interpolation, or extrapolation. Usually, a synthetic instance is produced by two or more randomly selected from the minor class so that its corresponding label becomes the same as the minor class. However, Krawczyk [2016] remarked that ROS methods, including SMOTE, do little to help improve the classifier's performance in extremely unbalanced problems. Along with Krawczyk [2016], our research suggested that generating synthetic examples based on random samples from only one class didn't help much in solving the imbalanced problem.

The other way to solve an extremely imbalanced problem is anomaly detection. Anomaly detection problems are generally aimed at classifying whether new test samples are normal or abnormal based on an anomaly score, assuming that a model trained using samples only from the normal class [Chalapathy and Chawla, 2019; Chandola et al., 2009]. For example, one-class classification finds a hypersphere that contains most of the normal data and predicts samples that fall outside the hypersphere as abnormal data [Ruff et al., 2018; Schölkopf et al., 2001; Tax and Duin, 2004]. In contrast to the standard anomaly detection setup, supervised anomaly detection using abnormal samples during the training phase has been studied [Görnitz et al., 2013; Liu and Zheng, 2006; Ruff et al., 2019]. These methods can be easily applied to imbalance problems by considering normal as a major class and abnormal as a minor class.

In this work, we proposed two novel methods for the imbalanced problem: (1) a new data-augmentation method inspired by mixup [Zhang et al., 2018]; (2) a new algorithm method that transforms the anomaly detection algorithm. Mixup is originally a regularization technique that trains a neural network using pairs of features and label interpolations. There are considerations for adapting the mixup to an imbalanced problem. First, since classes are extremely imbalanced, interpolations in which randomly selected pairs of examples are concentrated on major classes; thus, the mixup might have an adverse effect on a learning problem. Also, to mitigate the imbalance in the data set, each support of minor classes needs to be expanded to learn underlying decision boundaries. According to these considerations, we suggest a novel ROS method conducive to learning decision boundaries when classes are imbalanced and simple as the mixup.

Meanwhile, we observed that most imbalanced methods fail to learn an accurate model in extremely imbalanced settings. Hence, extending anomaly detection, we develop an algorithm to find compact hyperspheres that contain samples from the major class and extrude samples from the minor class. We can find compact hyperspheres by adding a penalty term that pushes minor samples to the objective function at the center of a hypersphere.

We briefly review related works in Section 3.2. We proposed our data augmentation method, supervised anomaly detection algorithm in Sections 3.3 and 3.4, respectively. To verify our method, Section 3.4 represents simulation studies of several related works and compares prediction powers of major, minor classes for each method with a certain level of imbalance. Especially, we conduct experiments by varying levels of imbalance for each method to perform comparative analysis.

3.2 Related works

3.2.1 Imbalanced classification

Data level methods modified training distribution using a sampling approach to alleviate class imbalance. It includes random oversampling (ROS), random undersampling (RUS). ROS duplicates random samples from minor classes, and RUS discards random samples from major classes [Barandela et al., 2004; He and Garcia, 2009; He and Ma, 2013; Van Hulse et al., 2007]. In general, ROS requires more computations because of duplications; meanwhile, RUS may lose some information by reducing the training dataset.

Oversampling methods can be divided into three different ways: simple, interpolation, and extrapolation. First, the simple repetition method is that random samples from a minor class are duplicated. Combining k-nearest neighbors and interpolation approach is known as SMOTE (Synthetic Minority Over-sampling TEchnique, Chawla et al. [2002]). SMOTE is an over-sampling method that the minority class is over-sampled by creating synthetic examples for imbalanced classification problems. Further, Ando and Huang [2017] suggested random-oversampling in an embedding space based on k-NN. Also, Nickerson et al. [2001] proposed an oversampling process combined with clustering. Before training a classifier, a clustering method such as k-means algorithm is fitted to be used to identify under-represented clusters with respect to the largest cluster. Then re-sampling approach is applied.

Several RUS methods have been proposed aimed at reducing the examples of major classes. Wilson [1972] suggested that first train a k-Nearest Neighbors (NN) classifier without labels, and then discard examples where that class label is not the same as the class label that references the most labels within the nearest neighbor. Also, Kubat et al. [1997] suggested that discarding major class examples that are 'redundant' or positioned at a 'border' line between major and minor classes treated as a noisy example. After that, Mani and Zhang [2003] chose the major class examples which closed to minor class examples based on k-NN, Drown et al. [2009] adopt the genetic algorithm to develop an RUS method. RUS methods are very attractive in terms of computation but hardly used for training deep neural networks because of DNNs complexity.

Moreover, Debowski et al. [2012] proposed dynamic sampling, which iteratively conducts RUS and ROS based on a pre-defined performance metric of the validation set. Lee et al. [2016] suggested the two-phase learning method in which a model is first pre-trained with threshold data, then fine-tuned using all data. Threshold data are generated by duplicating through noise injection or simple augmentations, or RUS. Koziarski [2020]; Koziarski et al. [2017] proposed radial-based oversampling and undersampling, used a radial-based function to estimate the mutual class distribution, and figured out regions where oversampling or undersampling should be applied. Data-level methods are usually applied before training as the data-preprocessing procedure, and then based on an artificial class-balanced dataset, a model is trained. Further, we refer Buda et al. [2018]; He and Garcia [2009]; Johnson and Khoshgoftaar [2019] to readers for additional data-level and algorithm-level methods.

On the other hand, algorithm-level methods for an imbalanced problem either propose a new loss function that gives more weight to the minor class or develop class weights that cost differently for misclassified examples of different labels or adapt an output in accordance with costs [Buda et al., 2018; Domingos, 1999; Elkan, 2001; Kukar et al., 1998; Zhou and Liu, 2005]. Further, to impose more weight on minor classes, weighted cross-entropy loss functions are proposed by Wang et al. [2016]. As a special case of weighted loss functions, Wang et al. [2016] suggested Mean False Error and its variation, which are based on the average of the mean squared errors calculated individually in different classes.

Also, in the object detection problem, Lin et al. [2017] proposed a loss function called focal loss (FL), which reshapes the cross-entropy loss to reduce the impact that easily classified samples in total loss. Basically, focal loss down-weights the loss assigned to well-classified examples and up-weights the loss assigned to poorly classified examples. After that, Cui et al. [2019] proposed class weighted loss function using the expected volume of sample for each class. Xu et al. [2020] proposed the worst case of weighted risk for imbalanced problem over a set of possible weights, and robust class weighted risk suggested including its generalization risk bound and optimization.

Moreover, based on hard mining [Rowley et al., 1998], class rectification loss (CRL) is proposed by Dong et al. [2017]. For every minor example in a mini-batch, CRL first searched for hard examples, each with the same/different labels respectively, and then learned decision boundaries so that, in a latent space, hard examples with the same labels are close to each other and hard examples with different labels are farther apart. Also, Huang et al. [2016, 2019] suggested methods to learn discriminatory deep representations via quintuple sampling or clustering and loss functions. After learning deep representation, for classification, they used a k-NN classifier based on learned representation. Another approach for cost-sensitive learning is to learn data-adaptive costs in the training stage without fixing weights in advance [Chung et al., 2016; Khan et al., 2017].

3.2.2 Mixup

Zhang et al. [2018] proposed a data-augmentation method, so-called mixup. Mixup regularizes a neural network using convex combinations of pairs of examples and their labels. Simply, mixup constructs virtual training examples as follows,

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \ \text{ where } \mathbf{x}_i,\, \mathbf{x}_j \text{ are raw input vectors} \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j, \ \text{ where } y_i,\, y_j \text{ are one-hot label encodings} \end{aligned} \tag{3.1}$$

where $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ are two examples drawn at random, and $\lambda \in [0, 1]$. To generating λ for a mixup coefficient, Zhang et al. [2018] suggested the beta distribution, Beta(α, α) with $\alpha > 0$. Mixup is simple and improves generalization in balanced classification problems. However, in extremely imbalanced classification, we show that simple mixup does not improve prediction performance in minor class, and further, we propose a data-augmentation method for class-imbalance problems based on the mixup.

3.2.3 Anomaly detection

Literature studies related to anomaly detection are extensive and are beyond the scope of this thesis, so we refer to Chalapathy and Chawla [2019]; Chandola et al. [2009] for more comprehensive studies. This thesis focused on supervised anomaly detection that can be extended/applied to extremely imbalanced problems. Further, in order to consider anomaly detection in imbalanced classification problems, normal data is considered to be a sample of the major class, and abnormal data is considered to be a sample of a minor class.

First, the one-class classification method finds the smallest hypersphere with a center and radius and detects an anomaly if it leaves the hypersphere [Ruff et al., 2018; Schölkopf et al., 2001; Tax and Duin, 2004], or uses reconstruction errors as an anomaly score and detects an anomaly if it has a higher anomaly score than a certain threshold [An and Cho, 2015; Chen et al., 2017; Hawkins et al., 2002; Sakurada and Yairi, 2014]. Also, using generative models such as kernel density estimation or deep generative models, Breunig et al. [2000]; Zhai et al. [2016]; Zong et al. [2018] proposed methods of estimating the distribution with high probability for the training (normal) data and classifying low-density examples as ideal in the test phase.

Unlike the one-class classification, which uses only normal data in the training phase, supervised anomaly detection (SAD) assumes some abnormal data in the training set. Görnitz et al. [2013]; Liu and Zheng [2006]; Ruff et al. [2019] proposed a methodology for estimating a compact hypersphere of normal data with center and radius by penalizing abnormal data to deviate from its center. In fact, semi-supervised settings are used in Görnitz et al. [2013]; Ruff et al. [2019]. They assumed training data composed of unlabeled data and labeled data and placed unlabeled data within (slightly off) the hypersphere of normal data.

3.3 Proposed methods

Notations and assumptions

We consider a binary-class classification problem. First, we denote $(\mathbf{x}_i, y_i)$ as the i-th pair of dataset where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is a p-dimensional feature vector and $y_i \in \{0, 1\}$ is the corresponding label where $y = 0$ denotes a sample from major class and $y = 1$ denotes a sample from minor class. Let $\mathcal{D}_M = \{(\mathbf{x}_i, 0)\}_{i=1}^{N}$ be the dataset from major classes and $\mathcal{D}_m = \{(\mathbf{x}_i, 1)\}_{i=1}^{n}$ be the dataset from minor classes, and N, n are the numbers of samples from major/minor classes respectively. To consider class-imbalance, we assume $N \gg n$. A model parameterized by θ is denoted as $f_\theta : \mathcal{X} \to [0,1]^C$.

3.3.1 MixupROS

In extremely imbalanced problems, it is essential to identify support for each minor class. Our framework, named *MixupROS*, tries to generate virtual examples close to minor classes so that supports could be expanded. To implement this, we generate a synthetic dataset combining ROS with mixup. Mixup generates virtual examples between two classes in a mini-batch. Hence, it enforces to efficiently determine decision boundaries between classes. However, it can be difficult for extremely imbalanced cases to include both major and minor class examples in one mini-batch. Even if some mini-batches contained both major and minor class examples, a mixup is done, emphasizing examples of major classes.

The difficulties of ROS methods are illustrated in Figure 3.1. The following experiment was performed as a toy example to compare how well supports of minor classes are extended. We generate major class examples from $\mathcal{N}([1,1]^\top, [0.5, 0; 0, 0.5])$ and minor class examples from $\mathcal{N}([-2,-2]^\top, [0.1, 0; 0, 0.1])$, and set $N = 100$, $n = 5$. Also, we set a over-sampling size as $\lfloor \frac{N-n}{2} \rfloor$. We note that in this example, the level of imbalance is $20 : 1$, which is not an extremely imbalanced case but gives intuitive and straightforward results.

Figure 3.1 represents examples with the major and minor class

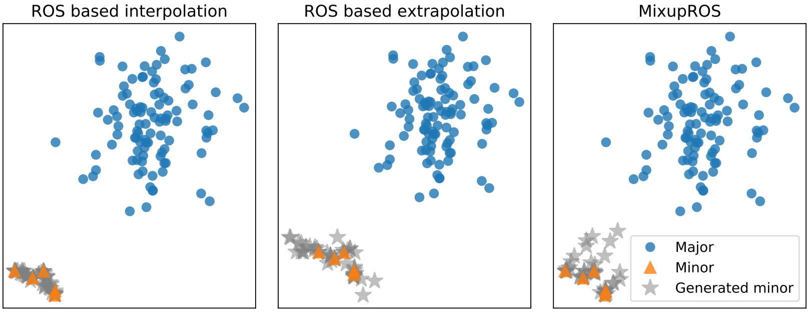

Figure 3.1: Results of data-augmentation methods for toy example. Blue circles/orange triangles indicate subjects from major/minor classes, respectively, and gray stars indicate augmented data which is used in the training phase.

as a blue circle, orange triangle, and augmented dataset, which will be used in the training phase as a gray star. We observe ROS-based interpolation and extrapolation produced augmented data only around the minor class. On the other hand, our proposed method, MixupROS, generates augmented data which are located between classes and close to minor class.

Our oversampling process pre-generates mixup data before training to generate a sufficiently large amount of synthetic minor class examples. Also, to give more weight to the minor class when performing mixup, a generalized version of the mixup was used. A generalized version of mixup changes the beta distribution sampling for the mixup coefficient λ, from $\text{Beta}(\alpha, \alpha)$ to $\text{Beta}(\alpha, \beta)$ where $\alpha, \beta > 0$. For example, suppose that $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ are random samples in training data. If one sample is from major and the other sample is from minor class, then mixup is applied with λ generated from $\text{Beta}(1,5)$ or $\text{Beta}(5,1)$ so that the weight of the minor class is larger than that of major class.

For a over-sampling size $k > 0$, MixupROS performs as the following steps:

1. Initialize a oversampling dataset $\mathcal{D}_{\text{gen}} = \emptyset$.

2. Generate over-sampling data using Mixup:

    - Randomly select k major samples without replacement, denote by $\tilde{\mathcal{D}}_M$, and k minor samples with replacement, denote by $\tilde{\mathcal{D}}_m$.

    - Generate k data as

    $$\begin{aligned}\mathcal{D}_{\text{gen}} = \mathcal{D}_{\text{gen}} \cup \{(\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \tilde{y} = 1) :\\ (\mathbf{x}_i, y_i) \in \tilde{\mathcal{D}}_M, (\mathbf{x}_j, y_j) \in \tilde{\mathcal{D}}_m\},\end{aligned}$$

    where $\lambda \sim \text{Beta}(\alpha, \beta)$ such that $\mathbb{E}\lambda$ is close to 0.

3. Train a model using $\mathcal{D}_M, \mathcal{D}_m, \mathcal{D}_{\text{gen}}$.

By oversampling using mixup in the proposed manner, we increase the number of examples with minor classes, thus provide more information to figure out supports of those. We note that MixupROS adopts mixup only on feature vectors to generate minor class examples. Our experimental studies suggest that the original mixup, which augments both feature and label, does not give another gain with respect to accuracy performance.

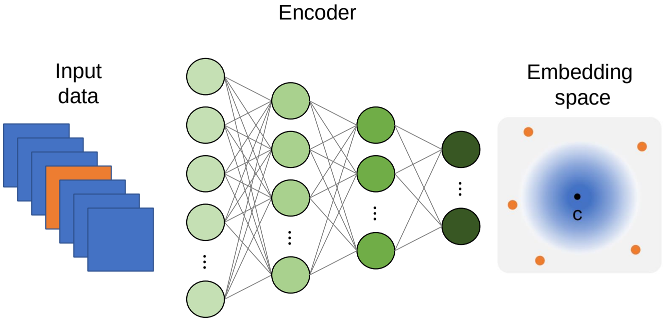

Figure 3.2: Framework of DeepSAD. In the latent space, the blue sphere represents the confidence region for major examples, and the orange points indicate the embedding vectors of minor examples.

3.3.2 Extensions of DeepSAD

Most imbalanced methods do not work well for extremely imbalanced classification problems. Hence, we present extensions of DeepSAD [Ruff et al., 2019], a supervised anomaly detection algorithm, to develop a new imbalanced classification algorithm in this section. DeepSAD aims to find an embedding mapping where the embedding vectors of the major class are near a predetermined center $\mathbf{c}$, and the embedding vectors of the minor class are far from the center (see Figure 3.2).

Let d be the dimension of embedding space, and let $\phi(\cdot; \Theta)$ be an embedding mapping from $\mathcal{X} \subset \mathbb{R}^p$ to embedding space $\mathcal{Z} \in \mathbb{R}^d$ with parameters Θ. Then the objective function of DeepSAD is to minimize

$$\frac{1}{N}\sum_{i\in\mathcal{D}_M}\|\phi(\mathbf{x}_i;\Theta)-\mathbf{c}\|^2+\frac{\lambda_1}{n}\sum_{j\in\mathcal{D}_m}\left[\|\phi(\mathbf{x}_j;\Theta)-\mathbf{c}\|\right]^{-1},$$

where $\lambda_1 > 0$ is a tuning parameter.

Meanwhile, we have observed that some embeddings of the minor class are clustered as the left panel in Figure 3.3. Ideally, the embedding vector of the minor examples should surround a hypersphere of the major example embedding vectors, but it can form clustering and degrade the performance of the classification model. Hence, we consider an additional penalty function for moving each embedding in the minor class away from the others, so that maximize the effectiveness of DeepSAD (see the right panel in Figure 3.3).

Motived by this observation, we propose the objective function for extensions of DeepSAD as follows:

$$\begin{aligned}&\frac{1}{N}\sum_{i\in\mathcal{D}_M}\|\phi(\mathbf{x}_i;\Theta)-\mathbf{c}\|^2+\frac{\lambda_1}{n}\sum_{j\in\mathcal{D}_m}\left[\|\phi(\mathbf{x}_j;\Theta)-\mathbf{c}\|\right]^{-1}\\&+\frac{\lambda_2}{|\{j,k:j,k\in\mathcal{D}_m,j\neq k\}|}\sum_{\substack{j,k\in\mathcal{D}_m,\\ j\neq k}}\left[d(\phi(\mathbf{x}_j;\Theta),\phi(\mathbf{x}_k;\Theta))\right]^{-1},\end{aligned}\tag{3.2}$$

where λ_1, λ_2 are positive tuning parameters and $d(\cdot,\cdot)$ is a metric in the embedding space. Here, λ_1 controls how far away the embedding vectors of minor examples should be from the center of the hypersphere, and λ_2 controls how far the embedding vectors of minor examples are from each other. In this thesis, we consider $d(\cdot,\cdot)$ as Euclidean norm or the cosine dissimilarity.

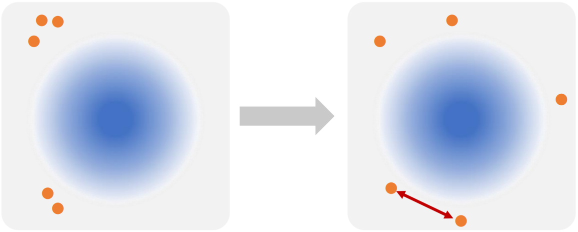

Figure 3.3: The motivation for extensions of DeepSAD.

3.4 Experiments

To evaluate proposed algorithms, we used the following five datasets listed in Table 3.1. For CIFAR-10, we randomly selected two classes among ten classes, set one of them as the major class and the other as the minor class, and repeated this procedure 30 times. As shown in Table 3.1, the class imbalance in the dataset is generally not very severe. After training/validation/test dataset split, we randomly down-sampled minor samples (n) in the training dataset for extremely imbalanced problem settings. In this experiments, we consider $n \in \{20, 10, 5\}$.

We used standard MLP networks on anomaly detection benchmark datasets and LeNet-type convolutional neural networks (CNNs) on CIFAR-10. The model is trained based on logistic loss and evaluated with the area of the ROC curve (AUC) and sensitivity with a certain level of specificity. For data-augmentation methods, we considered random oversampling with interpolation (ROS) [Chawla et al., 2002], Mixup [Zhang et al., 2018] and MixupROS where oversampling size is $(N - n)/2$. In MixupROS, oversampling

Dataset	N	n	p	Imbalance ratio
Arrhythmia	386	66	274	6:1
Cardio	1,655	176	21	9:1
Satellite	4,399	2,036	36	2:1
Speech	3,625	61	400	59:1
CIFAR-10	10,000	10,000	3,072	1:1

Table 3.1: Description of datasets.

data is generated using half interpolation and half MixupROS. Interpolation parameter λ is generated from beta distribution $\mathcal{B}(1,1)$ for ROS and Mixup, and for MixupROS λ is generated from beta distribution $\mathcal{B}(\alpha,\beta)$, and tuning parameters α, β are selected based on the validation data AUC among the following candidate parameter sets: $\alpha \in \{0.01, 0.05, 0.1, 1\}$ and $\beta \in \{0.5, 1, 5, 10, 20, 50\}$.

For one-class classification, we considered one-class Deep SVDD (DeepSVDD) [Ruff et al., 2018]. Also, DeepSAD [Ruff et al., 2019] which is a semi-supervised anomaly detection method, is applied for imbalanced classification. DeepSVDD, DeepSAD, and the proposed methods need to predetermine a center $\mathbf{c} \in \mathbb{R}^d$ in the embedding space, so we first train the autoencoder model using only major samples. Then we estimate $\mathbf{c}$ as the mean of major class embedding vectors and train a classifier by separating only the encoder model from the autoencoder model. We note that the weight parameters of the classifier models are initialized using the weights of the encoder model. Also, we fix the tuning parameter for DeepSAD and its extensions, λ_1, to 1, suggested by the authors of Ruff et al. [2019]. Finally, for the additional penalty in the extension of DeepSAD, the weighted average between Euclidean distance and cosine similarity is considered, and its tuning parameter is chosen based on the validation data AUC.

Results. Experimental results are shown in Table 3.2-3.6, with varying the number of minor samples denoted by n. First, we compare the data-augmentation methods. As the number of minor samples in the training data decreases, Supervised and Mixup, those that do not target the imbalance problem, suffer significant performance degradation. Meanwhile, ROS and MixupROS (proposed), Mixup with directional extrapolation, help learn better classifiers. We observe that among the data augmentation methods, MixupROS performs well in all cases, especially it performs better than other augmentation methods when the number of minor samples is small.

On the other hand, we observe that our proposed algorithm (DeepSAD ext.) performs best in all cases, in the deep anomaly detection methods. DeepSVDD, which trains using only information from major samples, has the lowest performance. Meanwhile, DeepSAD and its extension utilize minor samples to train classifiers with better performance. These numerical studies also show that we train a classifier with a better predictive performance by placing the embedding vectors of minor samples farther away than DeepSAD.

3.4.1 Comparison of distance functions in the extension of DeepSAD

Further, we analyze the sensitivity of the choice of distance function by comparing the extension of DeepSAD using various distance functions. Euclidean distance can be made large simply by placing the embedding vector of the main sample away from the center of the embedding space. Hence, maximizing the Euclidean distance may not mean finding an embedding mapping where minor samples can enclose the hypersphere of the major example embedding vectors. Therefore, in this experiment, Euclidean distance, cosine similarity, and their weighted sum distance are considered. Also, we fix the tuning parameter λ_1 as 1. Figure 3.4 shows that using the weighted average distance between Euclidean distance and cosine similarity gives a classifier the best test AUC.

Table 3.2: Results of arrhythmia dataset with varying n, where n is the number of minor samples in training set, and its proportions is given. Average (standard deviation) performances over 10 seeds are described. Items in bold indicates the best method.

		Baseline	Data augmentation			One-class Classification	Supervised AD	
n (ratio)	metric	Supervised Classifier	ROS	Mixup	Mixup-ROS	Deep-SVDD	DeepSAD	DeepSAD ext.
20 (8.0 %)	AUC	76.1 (4.7)	76.0 (5.6)	78.4 (4.3)	77.9 (5.5)	78.2 (3.9)	84.5 (2.6)	**85.9 (4.2)**
	Sens | Spec=80%	66.7 (6.3)	66.7 (9.6)	67.0 (5.9)	67.0 (7.9)	58.9 (8.3)	75.6 (7.0)	75.2 (11.5)
	Sens | Spec=90%	54.4 (9.2)	53.3 (10.1)	57.8 (11.5)	52.6 (9.5)	43.0 (7.0)	61.5 (12.9)	63.3 (13.3)
	Sens | Spec=95%	48.1 (13.2)	44.8 (11.6)	46.7 (10.1)	45.9 (9.8)	31.5 (8.2)	45.9 (17.1)	49.6 (16.4)
10 (4.1 %)	AUC	66.9 (8.0)	67.2 (8.7)	67.2 (9.1)	69.6 (9.3)	78.2 (3.9)	83.0 (2.9)	**83.8 (4.4)**
	Sens | Spec=80%	56.7 (9.6)	56.3 (9.2)	54.4 (11.8)	60.0 (10.6)	58.9 (8.3)	71.9 (6.1)	75.6 (8.6)
	Sens | Spec=90%	46.7 (9.1)	45.2 (9.4)	47.0 (10.5)	46.7 (11.7)	43.0 (7.0)	57.4 (8.4)	57.8 (12.1)
	Sens | Spec=95%	34.4 (6.8)	33.3 (7.2)	37.8 (9.4)	37.8 (11.4)	31.5 (8.2)	43.3 (12.7)	43.0 (11.2)
5 (2.1 %)	AUC	57.7 (9.6)	58.9 (10.2)	57.8 (8.4)	61.7 (8.5)	78.2 (3.9)	80.6 (3.7)	**81.9 (2.9)**
	Sens | Spec=80%	43.3 (10.8)	43.7 (12.9)	43.3 (11.0)	50.4 (11.2)	58.9 (8.3)	66.7 (7.0)	69.6 (8.5)
	Sens | Spec=90%	33.7 (11.0)	36.7 (13.3)	33.7 (10.4)	41.9 (11.7)	43.0 (7.0)	49.6 (10.1)	52.2 (7.1)
	Sens | Spec=95%	27.8 (10.4)	28.5 (10.9)	28.1 (11.7)	33.3 (10.6)	31.5 (8.2)	37.0 (8.0)	33.7 (5.4)

Table 3.3: Results of cardio dataset with varying n, where n is the number of minor samples in training set, and its proportions is given.. Average (standard deviation) performances over 10 seeds are described. Items in bold indicates the best method.

		Baseline	Data augmentation			One-class Classification	Supervised AD	
n (ratio)	metric	Supervised Classifier	ROS	Mixup	Mixup-ROS	Deep-SVDD	DeepSAD	DeepSAD ext.
20 (2.0 %)	AUC	96.4 (1.4)	97.7 (1.6)	95.8 (1.7)	**98.2 (1.2)**	73.6 (5.4)	97.6 (1.4)	**98.2 (0.9)**
	Sens | Spec=80%	94.8 (3.4)	97.0 (2.2)	93.8 (3.3)	97.2 (1.9)	53.2 (5.9)	97.2 (3.0)	98.5 (1.9)
	Sens | Spec=90%	91.0 (4.2)	94.5 (2.4)	89.4 (5.4)	95.5 (3.0)	39.0 (10.3)	94.4 (4.0)	96.1 (4.0)
	Sens | Spec=95%	87.0 (4.6)	92.5 (3.3)	85.5 (6.4)	93.9 (2.9)	29.6 (10.0)	90.1 (4.9)	90.4 (4.2)
10 (1.0 %)	AUC	90.2 (6.0)	95.8 (3.6)	88.7 (5.8)	96.5 (2.1)	73.6 (5.4)	96.3 (1.3)	**97.7 (0.7)**
	Sens | Spec=80%	83.7 (8.7)	92.8 (8.5)	83.4 (8.6)	94.9 (1.9)	53.2 (5.9)	94.9 (2.5)	97.7 (2.4)
	Sens | Spec=90%	78.3 (11.6)	90.3 (9.4)	77.5 (10.3)	92.0 (6.3)	39.0 (10.3)	90.7 (3.9)	93.2 (3.1)
	Sens | Spec=95%	73.8 (13.1)	88.3 (11.3)	70.4 (11.4)	86.5 (11.7)	29.6 (10.0)	83.5 (6.3)	88.9 (3.7)
5 (0.5 %)	AUC	74.8 (12.7)	86.0 (10.6)	71.4 (13.9)	92.4 (3.7)	73.6 (5.4)	90.6 (4.0)	**94.3 (5.1)**
	Sens | Spec=80%	64.9 (17.3)	77.7 (17.5)	61.4 (17.3)	87.9 (6.6)	53.2 (5.9)	84.8 (8.9)	91.3 (10.1)
	Sens | Spec=90%	56.9 (17.0)	74.5 (16.9)	53.5 (17.1)	82.7 (10.4)	39.0 (10.3)	74.2 (10.3)	85.9 (13.8)
	Sens | Spec=95%	49.9 (16.7)	71.4 (16.4)	46.2 (14.3)	76.9 (10.8)	29.6 (10.0)	67.2 (9.1)	78.6 (14.2)

Table 3.4: Results of satellite dataset with varying n, where n is the number of minor samples in training set, and its proportions is given. Average (standard deviation) performances over 10 seeds are described. Items in bold indicates the best method.

		Baseline	Data augmentation			One-class Classification	Supervised AD	
n (ratio)	metric	Supervised Classifier	ROS	Mixup	Mixup-ROS	Deep-SVDD	DeepSAD	DeepSAD ext.
20 (0.7 %)	AUC	86.2 (3.7)	90.8 (1.5)	85.5 (4.1)	91.1 (1.2)	82.4 (3.2)	91.1 (1.3)	**91.9 (1.4)**
	Sens | Spec=80%	76.7 (7.1)	85.2 (2.6)	75.4 (8.0)	85.7 (2.0)	67.5 (5.8)	84.5 (2.9)	86.1 (3.7)
	Sens | Spec=90%	62.3 (8.5)	75.6 (2.1)	60.5 (9.0)	77.0 (1.6)	55.1 (5.4)	73.3 (3.3)	76.6 (4.1)
	Sens | Spec=95%	51.7 (7.1)	66.1 (1.6)	51.0 (8.1)	67.6 (1.6)	46.0 (4.3)	64.3 (3.4)	67.7 (2.9)
10 (0.4 %)	AUC	74.5 (11.8)	85.9 (10.4)	72.6 (13.0)	88.4 (3.2)	82.4 (3.2)	88.7 (2.5)	**90.6 (1.5)**
	Sens | Spec=80%	59.1 (16.3)	78.6 (12.4)	57.0 (18.4)	81.8 (4.6)	67.5 (5.8)	79.7 (5.3)	83.6 (3.4)
	Sens | Spec=90%	44.9 (15.7)	68.1 (12.7)	41.4 (17.3)	73.0 (3.8)	55.1 (5.4)	66.9 (5.8)	72.9 (3.3)
	Sens | Spec=95%	33.7 (15.2)	58.0 (12.6)	30.1 (17.2)	64.4 (3.9)	46.0 (4.3)	58.0 (6.4)	63.9 (3.2)
5 (0.2 %)	AUC	63.9 (11.6)	76.5 (17.4)	62.3 (12.0)	82.0 (11.3)	82.4 (3.2)	86.0 (2.3)	**88.8 (2.0)**
	Sens | Spec=80%	43.5 (15.7)	64.2 (20.5)	41.7 (15.6)	72.5 (13.6)	67.5 (5.8)	74.4 (5.0)	79.2 (4.1)
	Sens | Spec=90%	29.2 (15.2)	54.5 (20.5)	26.5 (14.2)	64.1 (16.2)	55.1 (5.4)	62.0 (5.5)	69.5 (3.7)
	Sens | Spec=95%	20.1 (13.5)	46.4 (20.5)	17.8 (12.4)	57.2 (16.6)	46.0 (4.3)	52.4 (5.6)	62.4 (4.8)

Table 3.5: Results of speech dataset with varying n, where n is the number of minor samples in training set, and its proportions is given. Average (standard deviation) performances over 10 seeds are described. Items in bold indicates the best method.

		Baseline	Data augmentation			One-class Classification	Supervised AD	
n (ratio)	metric	Supervised Classifier	ROS	Mixup	Mixup-ROS	Deep-SVDD	DeepSAD	DeepSAD ext.
20 (0.9 %)	AUC	71.7 (3.7)	69.4 (4.3)	73.4 (5.9)	73.2 (5.8)	55.7 (7.3)	64.7 (5.3)	**73.6 (6.6)**
	Sens | Spec=80%	54.4 (7.6)	52.4 (6.9)	55.2 (8.2)	52.4 (10.7)	26.8 (7.6)	37.6 (5.7)	56.0 (12.5)
	Sens | Spec=90%	41.2 (9.2)	42.4 (7.6)	44.0 (7.8)	42.0 (8.9)	16.8 (9.0)	24.0 (7.8)	43.6 (7.2)
	Sens | Spec=95%	34.4 (7.8)	34.8 (5.3)	32.4 (7.4)	27.2 (9.9)	8.4 (7.6)	18.0 (8.3)	36.4 (7.4)
10 (0.5 %)	AUC	65.0 (6.9)	63.5 (7.3)	66.7 (6.8)	67.7 (4.1)	55.7 (7.3)	61.0 (5.4)	**70.2 (5.6)**
	Sens | Spec=80%	45.6 (10.2)	45.6 (9.8)	44.0 (11.2)	44.8 (9.2)	26.8 (7.6)	30.0 (7.8)	48.4 (9.1)
	Sens | Spec=90%	34.8 (10.0)	30.4 (8.7)	33.2 (8.2)	33.2 (8.4)	16.8 (9.0)	17.6 (8.5)	38.4 (9.5)
	Sens | Spec=95%	24.4 (7.2)	22.0 (6.6)	26.4 (7.8)	25.2 (5.3)	8.4 (7.6)	12.0 (6.0)	32.4 (9.1)
5 (0.2 %)	AUC	59.5 (6.0)	58.4 (5.5)	60.6 (7.3)	61.9 (7.6)	55.7 (7.3)	59.4 (6.7)	**64.0 (4.4)**
	Sens | Spec=80%	37.6 (10.0)	34.8 (10.5)	36.4 (11.7)	37.6 (11.0)	26.8 (7.6)	32.0 (13.7)	38.0 (10.2)
	Sens | Spec=90%	25.2 (10.0)	23.6 (10.4)	28.4 (9.9)	25.6 (10.0)	16.8 (9.0)	17.6 (6.3)	23.2 (7.7)
	Sens | Spec=95%	16.4 (8.5)	16.0 (6.3)	18.0 (7.4)	20.0 (8.4)	8.4 (7.6)	10.4 (3.4)	16.4 (5.5)

Table 3.6: Results of CIFAR10 dataset with varying n, where n is the number of minor samples in training set, and its proportions is given. Average (standard deviation) performances over 5 seeds are described. Items in bold indicates the best method.

		Baseline	Data augmentation			One-class Classification	Supervised AD	
n (ratio)	metric	Supervised Classifier	ROS	Mixup	Mixup-ROS	Deep-SVDD	DeepSAD	DeepSAD ext.
20 (0.4 %)	AUC	82.0 (11.3)	80.0 (12.4)	77.6 (11.7)	83.0 (11.1)	66.8 (13.9)	73.7 (12.5)	**86.1 (10.9)**
	Sens | Spec=80%	69.7 (19.5)	66.8 (20.9)	63.1 (19.5)	71.4 (19.4)	44.2 (18.2)	54.3 (19.0)	77.7 (18.9)
	Sens | Spec=90%	57.7 (22.7)	54.6 (23.4)	50.8 (21.3)	59.6 (22.6)	28.4 (14.4)	38.9 (17.7)	67.4 (22.7)
	Sens | Spec=95%	46.9 (23.2)	44.6 (24.0)	40.8 (20.5)	49.9 (23.5)	16.6 (9.4)	26.7 (14.9)	57.0 (24.2)
10 (0.2 %)	AUC	78.2 (12.5)	76.9 (13.9)	73.9 (12.6)	79.5 (11.8)	66.8 (13.9)	71.0 (13.0)	**81.5 (12.1)**
	Sens | Spec=80%	63.6 (21.3)	61.5 (22.6)	56.9 (20.5)	65.8 (20.0)	44.2 (18.2)	50.3 (19.2)	68.7 (20.6)
	Sens | Spec=90%	50.7 (22.5)	49.3 (23.6)	44.3 (20.4)	53.9 (22.2)	28.4 (14.4)	34.4 (16.6)	57.3 (22.7)
	Sens | Spec=95%	40.1 (21.3)	39.3 (22.9)	33.2 (18.1)	43.3 (22.5)	16.6 (9.4)	22.5 (13.1)	46.9 (23.6)
5 (0.1 %)	AUC	73.8 (13.0)	73.7 (14.2)	70.3 (12.9)	**75.0 (12.6)**	66.8 (13.9)	68.2 (13.6)	72.8 (12.3)
	Sens | Spec=80%	56.3 (20.8)	56.9 (22.2)	51.2 (19.2)	58.5 (20.6)	44.2 (18.2)	45.7 (18.7)	53.0 (18.7)
	Sens | Spec=90%	43.0 (20.2)	43.8 (21.8)	38.0 (18.2)	45.7 (21.5)	28.4 (14.4)	29.6 (14.8)	37.4 (17.5)
	Sens | Spec=95%	31.8 (17.2)	33.9 (20.0)	27.7 (15.7)	34.9 (20.2)	16.6 (9.4)	17.9 (10.6)	26.0 (14.9)

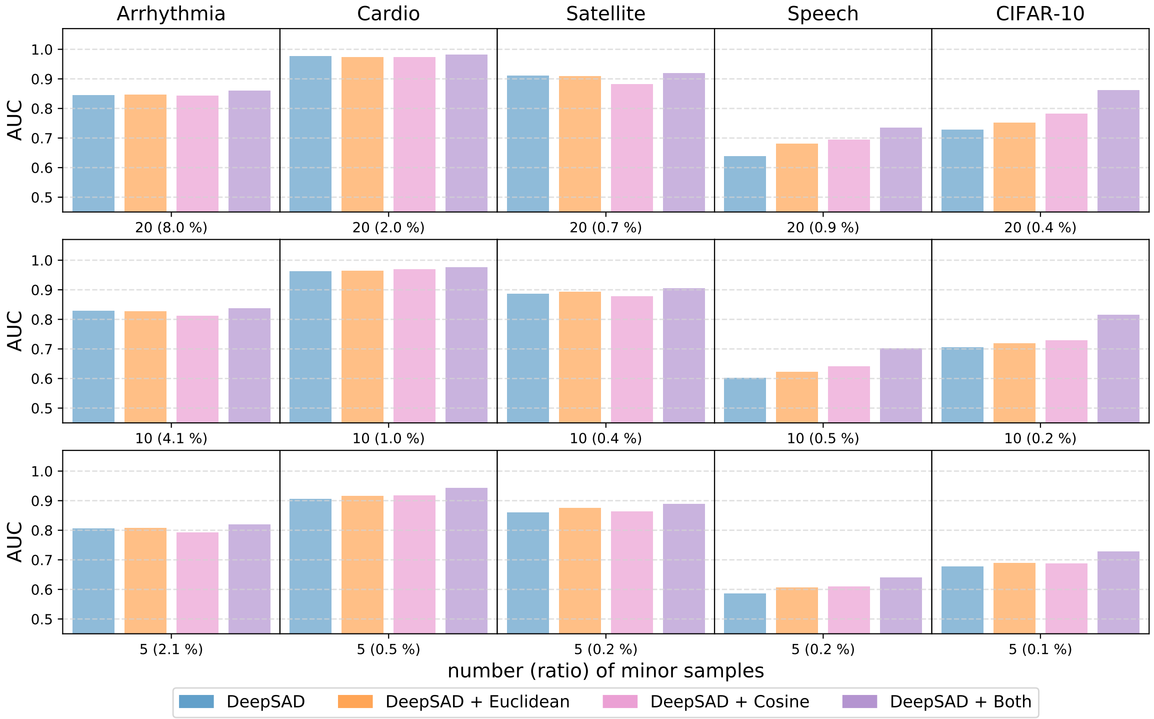

Figure 3.4: The extensions of DeepSAD sensitive analysis with respect to distance function. We summarize the average of the test AUC values varying the number of minor samples.

Chapter 4

Conclusions

In this thesis, we first introduced a better guide to BGF so-called *within-group fairness*, which should be considered along with BGF when fair AI is a concern. Also, we proposed a regularization procedure to control the degree of WGF of the estimated classifiers and score functions. By analyzing four real-world datasets, we illustrated that the WGF constraints improve the degree of WGF without hampering BGF as well as accuracy. Thus, we concluded that the doubly-fair algorithms find a fair AI model with respect to BGF and individual fairness in each sensitive group. Moreover, in many cases, the WGF constraints are helpful to find more accurate prediction models.

A problem in the proposed learning algorithm for WGF is that using a surrogated constraint for a given WGF constraint is sometimes problematic. The learning algorithm can find a DF model with a lower surrogated WGF value than a BGF model, but the original WGF value is much higher. See Section A.2 of Appendix for empirical evidence. A better-surrogated WGF constraint to ensure a lower original WGF value would be useful.

On the other hand, we proposed a new data-augmentation based on Mixup, named *MixupROS*, and an extension of DeepSAD for extremely imbalanced problems. MixupROS generates minor virtual examples between major and minor classes when oversampling minor samples. The extension of DeepSAD also moves each minor sample embeddings away from the other to surround the hypersphere of the major sample embedding while placing minor sample embeddings outside the hypersphere. Also, the extension of DeepSAD moves each minor sample embeddings away from the other to enclose a hypersphere of major sample embeddings. By considering the extremely imbalanced ratio for various datasets, our proposed algorithms show superior results compared to other works.

Bibliography

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 770–785. Springer, 2017.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23:2016, 2016.

Ricardo Barandela, Rosa M Valdovinos, J Salvador Sánchez, and Francesc J Ferri. The imbalanced training sample problem: Under or over sampling? In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pages 806–814. Springer, 2004.

Richard E Barlow. Statistical inference under order restrictions;

the theory and application of isotonic regression. Technical report, 1972.

Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001, 2017.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 319–328, 2019.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Philip K Chan, Wei Fan, Andreas L Prodromidis, and Salvatore J Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, 14(6):67–74, 1999.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58, 2009.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 90–98. SIAM, 2017.

Jaewoong Cho, Changho Suh, and Gyeongjo Hwang. A fair classifier using kernel density estimation. In *34th Conference on Neural Information Processing Systems, NeurIPS 2020.* Conference on Neural Information Processing Systems, 2020.

Yu-An Chung, Hsuan-Tien Lin, and Shao-Wen Yang. Cost-aware pre-training for multiclass cost-sensitive deep learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1411–1417, 2016.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Leveraging labeled and unlabeled data for consistent fair binary classification. In *Advances in Neural Information Processing Systems*, pages 12760–12770, 2019.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017.

Amanda Coston, Karthikeyan Natesan Ramamurthy, Dennis Wei, Kush R Varshney, Skyler Speakman, Zairah Mustahsan, and Supriyo Chakraborty. Fair transfer learning with missing protected attributes. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 91–98, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

Bazyli Debowski, Shawki Areibi, Gary Gréwal, and J Tempelman. A dynamic sampling framework for multi-class imbalanced data. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 113–118. IEEE, 2012.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, 1999.

Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1851–1860, 2017.

Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pages 2791–2801, 2018.

Dennis J Drown, Taghi M Khoshgoftaar, and Naeem Seliya. Evolutionary sampling and software quality modeling of high-assurance systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(5):1097–1107, 2009.

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

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.

Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

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.

Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 144–152. SIAM, 2016.

Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Vicente Garcia, J Salvador Sanchez, Ramon A Mollineda, Roberto Alejo, and Jose M Sotoca. The class imbalance problem in pattern classification and learning. In *II Congreso Español de Informática*, pages 283–291. Zarzgoza, 2007.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems*, pages 2415–2423, 2016.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

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

Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer, 2002.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9): 1263–1284, 2009.

Haibo He and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications.* John Wiley & Sons, 2013.

Paulina Hensman and David Masko. The impact of imbalanced training data for convolutional neural networks. *Degree Project in Co mputer Science, KTH Royal Institute of Technology*, 2015.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

David Ingold and Spencer Soper. Amazon doesn't consider the race of its customers. should it. *Bloomberg, April*, 1, 2016.

Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, pages 862–872. PMLR, 2020.

Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929. IEEE, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8): 3573–3587, 2017.

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27, 2018.

Michał Koziarski. Radial-based undersampling for imbalanced data classification. *Pattern Recognition*, 102:107262, 2020.

Michał Koziarski, Bartosz Krawczyk, and Michał Woźniak. Radial-based approach to imbalanced data oversampling. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 318–327. Springer, 2017.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Citeseer, 1997.

Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2-3):195–215, 1998.

Matjaz Kukar, Igor Kononenko, et al. Cost-sensitive learning with neural networks. In *ECAI*, volume 98, pages 445–449, 1998.

Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. ifair: Learning individually fair data representations for algorithmic decision making. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1334–1345. IEEE, 2019.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. Fairness without demographics through adversarially reweighted learning. *arXiv preprint arXiv:2006.13114*, 2020.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. *ProPublica (5 2016)*, 9(1), 2016.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pages 3713–3717. IEEE, 2016.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Yi Liu and Yuan F Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 129–132. IEEE, 2006.

Patrick Mair, Kurt Hornik, and Jan de Leeuw. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of statistical software*, 32(5):1–24, 2009.

Enno Mammen and Christine Thomas-Agnan. Smoothing splines and shape restrictions. *Scandinavian Journal of Statistics*, 26 (2):239–252, 1999.

Inderjeet Mani and I Zhang. kNN approach to unbalanced data distributions: a case study involving information extraction. In

Proceedings of workshop on learning from imbalanced datasets, volume 126, 2003.

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.

Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118, 2018.

Ofir Nachum and Heinrich Jiang. Group-based fair learning leads to counter-intuitive predictions. *arXiv preprint arXiv:1910.02097*, 2019.

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

Adam Nickerson, Nathalie Japkowicz, and Evangelos E Milios. Using Unsupervised Learning to Guide Resampling in Imbalanced Data Sets. In *AISTATS*, 2001.

Manisha Padala and Sujit Gujar. FNNC: Achieving Fairness through Neural Networks. pages 2249–2255, 07 2020. doi: 10.24963/ijcai.2020/311.

Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017.

Natalya Pya and Simon N Wood. Shape constrained additive models. *Statistics and Computing*, 25(3):543–559, 2015.

Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

Johannes Schmidt-Hieber et al. Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2239–2248, 2018.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

Sofia Visa and Anca Ralescu. Issues in mining imbalanced data sets-a review paper. In *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*, volume 2005, pages 67–73. sn, 2005.

Robin Vogel, Aurélien Bellet, and Stéphan Clémençon. Learning Fair Scoring Functions: Fairness Definitions, Algorithms and

Generalization Bounds for Bipartite Ranking. *arXiv preprint arXiv:2002.08159*, 2020.

Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pages 4368–4374. IEEE, 2016.

Xiao Wang and Feng Li. Isotonic smoothing spline regression. *Journal of Computational and Graphical Statistics*, 17(1):21–37, 2008.

Ehsan Ullah Warriach and Kenji Tei. Fault detection in wireless sensor networks: A machine learning approach. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 758–765. IEEE, 2013.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617, 2018.

Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio Calmon. Optimized Score Transformation for Fair Classification. volume 108 of *Proceedings of Machine Learning Research*, pages 1673–1683, Online, 26–28 Aug 2020. PMLR. URL `http://proceedings.mlr.press/v108/wei20a.html`.

Linda F Wightman and Henry Ramsey. *LSAC national longitudinal bar passage study*. Law School Admission Council, 1998.

Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

Yongkai Wu, Lu Zhang, and Xintao Wu. Fairness-aware Classification: Criterion, Convexity, and Bounds. *arXiv preprint arXiv:1809.04737*, 2018.

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

Ziyu Xu, Chen Dan, Justin Khim, and Pradeep Ravikumar. Class-weighted classification: Trade-offs and robust approaches. *arXiv preprint arXiv:2005.12914*, 2020.

Gal Yona and Guy Rothblum. Probably approximately metric-fair learning. In *International Conference on Machine Learning*, pages 5680–5688. PMLR, 2018.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

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.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, pages 1100–1109. PMLR, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*, 2018.

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1): 63–77, 2005.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

Appendix A

Appendix for Within-group fairness

A.1 Additional numerical studies for WGF classification

A.1.1 Targeting for disparate impact

First, we investigate the sensitivity of the prediction accuracy to the degree of dWGF in the DNN model. Figure A.1 shows the scatter plots between various dWGF values and the corresponding accuracies for the DF DNN model, where the DI is fixed around 0.03. The accuracies are not very sensitive to the dWGF values like the DF linear logistic model. Furthermore, for the datasets *Adult*, *Bank* and *COMPAS*, the DF classifiers have higher accuracies and lower dWGF values than the BGF classifier.

We also investigate how the dWGF constraint performs with surrogated BGF constraints other than Hinge-DI: (i) the covariance type constraint [Zafar et al., 2017, 2019], named by COV-DI; and (ii) the linear surrogated function, named by FNNC-DI [Padala and Gujar, 2020]. Table A.1 presents the results with various surrogated DI constraints and the dWGF constraint. In most cases, COV-DI and FNNC-DI give the results similar to Hinge-DI with or without the dWGF constraint and we consistently observe that considering the dWGF constraint together with the DI constraint helps to alleviate within-group fairness while maintaining similar levels of the accuracy and the DI. Note that for the dataset *Adult*, the DNN model with COV-DI constraint does not achieve the pre-specified DI value 0.03 regardless of the choice of tuning parameter. In contrast, the DNN model trained with the DI and dWGF constraints achieves the DI value 0.03 with a smaller value of dWGF. This observation is interesting since it implies that the dWGF constraint is helpful to increase even the BGF.

Next, we compare the dWGF and WGF constraints when targeting the DI with the hinge surrogated function in Table A.2. In most cases, both the dWGF and WGF constraints are helpful to improve the WGF, while maintaining a similar level of accuracy and DI. It is noticeable that the DF classifier with the dWGF constraint is more accurate than that with the WGF constraint, which would be mainly because the DI constraint is directional.

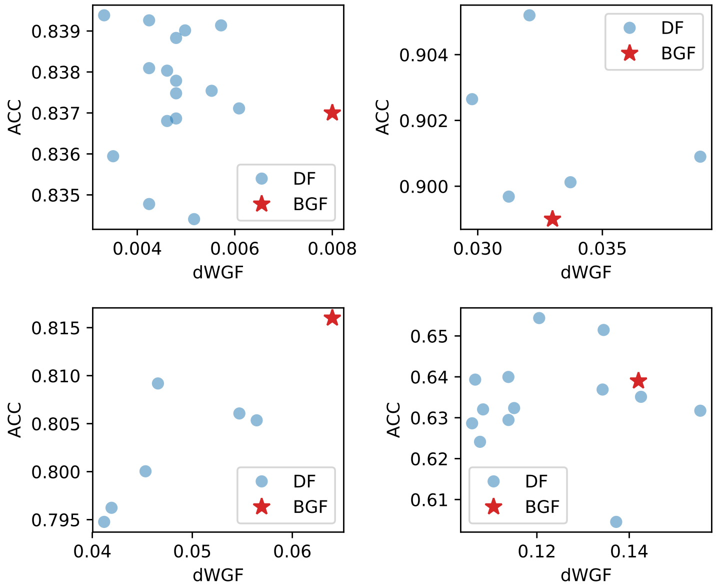

Figure A.1: Scatter plots of the accuracies and dWGF values for the DF DNN model with the DI values around 0.03. (Topleft) *Adult*; (Topright) *Bank*; (Bottomleft) *LSAC*; (Bottomright) *COMPAS*. Red star points in each figure represent the results of the BGF classifier.

Table A.1: Results for the DF classifier with various surrogated DI constraints. Except for the dataset *Adult*, average performances are described.

		Linear model			DNN model		
Dataset	Method	ACC	DI	dWGF	ACC	DI	dWGF
Adult	Uncons.	0.852	0.172	0.000	0.853	0.170	0.000
	COV-DI	0.837	0.035	0.003	0.845	0.082	0.013
	COV-DI-DF	0.837	0.030	0.001	0.840	0.025	0.007
	FNNC-DI	0.834	0.023	0.003	0.838	0.023	0.006
	FNNC-DI-DF	0.836	0.025	0.001	0.841	0.025	0.004
Bank	Uncons.	0.908	0.195	0.000	0.904	0.236	0.000
	COV-DI	0.904	0.019	0.009	0.906	0.019	0.036
	COV-DI-DF	0.904	0.020	0.007	0.906	0.020	0.033
	FNNC-DI	0.903	0.020	0.013	0.901	0.020	0.029
	FNNC-DI-DF	0.905	0.020	0.008	0.900	0.010	0.027
LSAC	Uncons.	0.823	0.120	0.000	0.856	0.131	0.000
	COV-DI	0.808	0.015	0.014	0.859	0.052	0.020
	COV-DI-DF	0.811	0.019	0.010	0.860	0.054	0.014
	FNNC-DI	0.809	0.020	0.014	0.851	0.025	0.023
	FNNC-DI-DF	0.809	0.014	0.010	0.844	0.010	0.019
COMPAS	Uncons.	0.757	0.164	0.000	0.757	0.162	0.000
	COV-DI	0.640	0.029	0.149	0.661	0.038	0.124
	COV-DI-DF	0.620	0.024	0.135	0.650	0.028	0.097
	FNNC-DI	0.646	0.037	0.146	0.646	0.032	0.133
	FNNC-DI-DF	0.624	0.034	0.143	0.645	0.021	0.117

Table A.2: Comparison of the dWGF and WGF constraints based on the linear logistic model. Except for the dataset *Adult*, average performances are described.

		with the dWGF constraint			with the WGF constraint		
Dataset	Method	ACC	DI	dWGF	ACC	DI	WGF
Adult	Hinge-DI	0.833	0.028	0.005	0.833	0.028	0.005
	Hinge-DI-DF	0.836	0.028	0.003	0.830	0.012	0.005
Bank	Hinge-DI	0.901	0.024	0.018	0.901	0.024	0.003
	Hinge-DI-DF	0.904	0.021	0.007	0.898	0.017	0.000
LSAC	Hinge-DI	0.809	0.017	0.014	0.809	0.017	0.014
	Hinge-DI-DF	0.813	0.018	0.009	0.810	0.016	0.011
COMPAS	Hinge-DI	0.641	0.024	0.153	0.641	0.024	0.136
	Hinge-DI-DF	0.618	0.025	0.145	0.594	0.018	0.088

A.1.2 Targeting for equal opportunity

We exam how the dWGF constraint works with the equal opportunity constraint given as

$$\mathrm{EOp} = |\Pr(\hat{Y} = 1|Y = 1, Z = 1) - \Pr(\hat{Y} = 1|Y = 1, Z = 0)|,$$

and the results are summarized in Table A.3. For some cases, the dWGF constraint does not work at all (i.e., the dWGF values of the BGF and DF classifiers are the sames). This is partly because the surrogated dWGF constraint does not represent the original dWGF well, which is discussed in the following section.

A.2 Limitations of surrogated WGF constraint

We have seen that the DF classifier does not improve the dWGF value at all compared to the BGF classifier with respect to the equal opportunity constraint for some datasets. We found that these undesirable results would be because the surrogated dWGF constraint using the hinge function does not represent the original dWGF constraint. To take a closer look at this problem, we investigate relations between the dWGF and W_{conv} evaluated on the training datasets *Bank* and *LSAC* in Figure A.2. We observe that the DF classifier has lower W_{conv} values but higher dWGF values than the BGF classifier. That is, reducing the W_{conv} value does not always result in a small value of the original dWGF. Alternative surrogated constraints, which resemble the original dWGF

Table A.3: Results for targeting EOp-dWGF. Except for the dataset *Adult*, average performances are described.

		Linear model			DNN model		
Dataset	Method	ACC	EOp	dWGF	ACC	EOp	dWGF
Adult	Uncons.	0.852	0.070	0.000	0.853	0.076	0.000
	Hinge-EOp	0.851	0.011	0.002	0.854	0.012	0.030
	Hinge-EOp-DF	0.853	0.016	0.001	0.854	0.015	0.012
	FNNC-EOp	0.851	0.013	0.012	0.852	0.004	0.021
	FNNC-EOp-DF	0.852	0.007	0.007	0.852	0.006	0.019
Bank	Uncons.	0.908	0.099	0.000	0.904	0.082	0.000
	Hinge-EOp	0.908	0.027	0.007	0.909	0.031	0.122
	Hinge-EOp-DF	0.908	0.027	0.007	0.909	0.031	0.122
	FNNC-EOp	0.908	0.027	0.010	0.903	0.037	0.111
	FNNC-EOp-DF	0.908	0.030	0.010	0.900	0.028	0.107
LSAC	Uncons.	0.823	0.041	0.000	0.856	0.038	0.000
	Hinge-EOp	0.820	0.003	0.004	0.852	0.010	0.015
	Hinge-EOp-DF	0.820	0.003	0.004	0.851	0.008	0.012
	FNNC-EOp	0.822	0.011	0.003	0.859	0.010	0.011
	FNNC-EOp-DF	0.822	0.011	0.003	0.858	0.010	0.010
COMPAS	Uncons.	0.757	0.074	0.000	0.757	0.075	0.000
	Hinge-EOp	0.713	0.042	0.073	0.719	0.029	0.046
	Hinge-EOp-DF	0.713	0.042	0.073	0.719	0.029	0.046
	FNNC-EOp	0.666	0.039	0.197	0.722	0.031	0.056
	FNNC-EOp-DF	0.706	0.031	0.092	0.725	0.035	0.042

closely but are yet computationally easy, are needed and we leave this issue for future work.

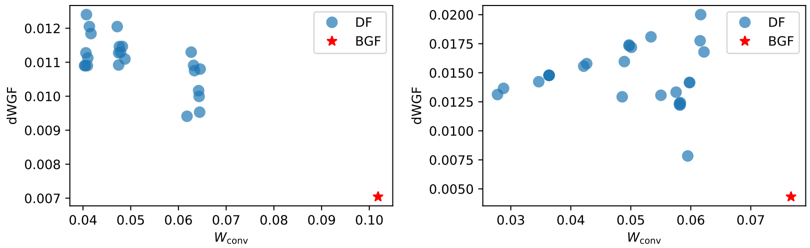

Figure A.2: Scatter plots of the dWGF and the within-group fairness penalty (W_{conv}) values for the DF linear logistic model with the EOp values around 0.03 evaluated on the training datasets. (Left) *Bank*; (Right) *LSAC*. Red star points in each figure represent the results of the BGF classifier.

A.3 Datasets and Preprocessing

Dataset. We conduct our experiments with four real-world datasets, which are popularly used in fairness AI research and publicly available:

- *Adult* [Dua and Graff, 2017]: The Adult Income dataset consists of 32,561 training subjects and 16,281 test subjects with 14 features and a binary target, which indicates whether income exceeds \$50k per a year. The sensitive variable is the sex of the subject, $Z = 0$ for female and $Z = 1$ for male.

- *Bank* [Dua and Graff, 2017]: The Bank Marketing dataset contains 41,188 subjects with 20 features (e.g. age, occupation, marital status) and a binary target indicating whether or not subjects have subscribed to the product (bank term deposit). A discrete age is set as a binary sensitive variable by assigning 0 to subjects aged 25 to 60 years old and 1 to else.

- *LSAC* [Wightman and Ramsey, 1998]: The Law School dataset pre-processed by [Lahoti et al., 2020] contains 26,551 subjects with 10 input variables and a binary target which indicates whether subject passed the bar exam or not. The sensitive variable is set by 0 for 'non-white' subjects and 1 for 'white' subjects.

- *COMPAS* [Larson et al., 2016]: The Compas Propublica Risk Assessment dataset contains 6,172 subjects to predict recidivism ('HighScore' or 'LowScore') with 6 variables related to criminal history and demographic information. We use racial characteristics as a sensitive variable.

We transform all categorical variables to dummy variables using one-hot encoding, and standardize to get zero mean and 1 standard deviation for each variable. Some variables having serious multicollinearity have been removed in order to obtain stable estimation results. The performances of the unconstrained linear logistic model are summarized in Table A.4.

Table A.4: Performances of the unconstrained linear logistic model on the test dataset. Except for *Adult*, average metrics are described.

Model	Dataset	Acc	DI	EOp	DM
Linear	*Adult*	0.852	0.172	0.070	0.117
	Bank	0.908	0.195	0.099	0.176
	LSAC	0.823	0.120	0.041	0.090
	COMPAS	0.757	0.164	0.074	0.020
DNN	*Adult*	0.853	0.170	0.076	0.105
	Bank	0.904	0.236	0.082	0.174
	LSAC	0.856	0.131	0.038	0.071
	COMPAS	0.757	0.162	0.075	0.024

A.4 Implementation details

For numerical stability, we use the ridge penalty for DNN parameters with the regularization parameter 10^{-6}. All experiments are conducted on a GPU server with NVIDIA TITAN Xp GPUs. Also, for each method, we consider lr $\in \{0.01, 0.1, 1\}$ and epoch $\in \{10000, 20000\}$, then we choose the best learning rate and epoch. In addition, we did not use a mini-batch for the gradient descent approach, i.e., we set the batch size to the sample size. For each BGF constraint, we choose the corresponding regularization parameter so that the value of the BGF constraint (e.g., DI, EOp, MSP) reaches a certain level among the following candidate parameters set:

$$\lambda \in \{0, 0.05, 0.1, 0.35, 0.45, 0.6, 0.75, 1, 2, 5\}.$$

The hyper-parameters in the doubly-fair algorithm are set to minimize the dWGF (or WGF) value while the BGF level remains similar to that of the BGF classifier, among the following candidate parameters sets:

$$\begin{aligned}
&\lambda \in \{0, 0.05, 0.1, 0.35, 0.45, 0.6, 0.75, 1, 2, 5\} \\
&\eta \in \{0, 0.1, 0.5, 1, 3, 5\}.
\end{aligned}$$

For the WGF score function, we adopt the surrogated version of Kendall's τ as the WGF constraint. However, the surrogated Kendall's τ requires huge computation since it should process all pairs of the training data. To save computing time for calculating the surrogated Kendall's τ, we use 50,000 pairs of samples randomly selected from the training data for each sensitive group.

국문초록

본 논문에서는 분류문제를 다룰 때 발생할 수 있는 두 가지 이슈에 대해 논의한다. 먼저 첫 번째 이슈는 인공지능의 공정성으로, 인공지능이 여러 분야에서 뛰어난 성능을 나타내어 사회적 의사결정 도구에 활용되면서, 인공지능은 정확하면서 특정 민감 그룹(예. 유색인종, 여성)에 대해 불공정을 내포해서는 안된다. 이를 해결하기 위해, 민감그룹 간 공정한 인공지능을 학습하는 다양한 알고리즘이 제안되었다. 한편, 그룹 간에 공정한 모형이 개개인을 불공정하게 대할 수 있는 문제점이 있어 개인간 공정성 개념이 제안되었지만, 이 개념은 개개인 간 유사도를 측정해야 하는 어려움 때문에 실생활에 적용하기 어렵다. 따라서 본 논문에서는 그룹 간 공정성을 학습할 때 그룹 내에서 불공정이 일어나지 않도록 하는 더 나은 방향의 가이드인 그룹 내 공정성을 소개한다. 그리고 그룹 내 공정성의 수학적 정의를 제안하여 그룹 내 공정성을 개념화하고, 이를 통제할 수 있는 알고리즘을 개발한다. 다양한 실험을 통해 제안한 알고리즘이 정확도와 그룹 간 공정성을 비슷하게 유지하면서 그룹 내 공정성을 완화시킴을 확인하였다.

둘째로, 클래스 간 자료의 수가 극단적으로 불균형할 때의 분류문제를 고려한다. 불균형 분류문제는 이상 거래 탐지, 의학 진단

등 다양한 분야에서 연구됐으며, 일반적으로 불균형 데이터셋으로 분류기를 학습할 경우 자료의 수가 더 많은 메이저 클래스에 초점을 맞춰 자료의 수가 적은 마이너 클래스의 특성을 잘 학습하지 못하게 된다. 이러한 불균형성을 해결하기 위해 가장 직관적이고 사용하기 쉬운 오버샘플링이 적용되었지만, 단순하게 마이너 클래스의 샘플을 복제하는 방법은 더 나은 분류기를 학습할 때 큰 도움이 되지 않는다. 이 논문에서는 새로운 데이터 증원법인 *MixupROS*와 딥러닝을 활용한 이상 탐지 방법의 확장 알고리즘을 제안한다. 극단적 불균형을 가정한 여러 데이터셋의 실험 결과를 통해, 제안하는 알고리즘이 기존의 방법들보다 우수한 성능을 가지는 분류기를 학습함을 확인하였다.

주요어: 분류모형, 분류문제, 공정한 인공지능, 그룹 내 공정성, 불균형 분류문제, 데이터 증원법, 지도 이상 탐지

학 번: 2014-22358