LEARNING WITH BIASED DATA: INVARIANT REPRESENTATIONS AND TARGET LABELS

THOMAS MAXIMILIAN KEHRENBERG

A thesis submitted for the degree of Doctor of Philosophy.
School of Engineering and Informatics
University of Sussex

March 2021

ABSTRACT

Biased data represents a significant challenge for the proper functioning of machine learning models, which affects the trustworthiness of deployed models. These biases are usually introduced by the data generation process, i.e., data is collected from non-representative samples or is the result of biased processes. However, these data deficiencies can be very expensive or even impossible to fix, which makes it desirable to solve the problem on the algorithmic end. In this work, I consider two different forms of data bias: labelling bias and sampling bias; investigated under the framework of algorithmic fairness and evaluated using common fairness metrics. Labelling bias here refers to a systematic bias, correlated with a sensitive attribute, which causes the labels in the dataset to differ from the "true" labels; whereas sampling bias indicates that samples are missing from the training set in a systematic way, but are still present in the setting where the model is intended to be deployed. Both biases will make a naively trained model fail to generalize. I present three approaches to tackling this problem, each relying on some form of additional knowledge about the data. The first approach, dealing with labelling bias, is based on implicit, probabilistic target labels which satisfy certain given statistics. These target labels can be used to train any likelihood-based model. The second approach deals with strong spurious correlations in the training data, which can be seen as a specific form of sampling bias. A bias-free partially-labelled context set is used to learn an interpretable representation of the data which is invariant to the spurious correlation and can be assessed qualitatively. The third approach deals with less extreme cases of sampling bias, but relaxes the assumption of having labels in the context set, by learning an invariant representation via distribution matching.

DECLARATION

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree. Except where indicated by specific stated in the text, this thesis was composed by myself and the work contained therein in my own.

Hove, March 2021

Thomas Maximilian Kehrenberg

ACKNOWLEDGEMENTS

I would like to thank my family for their support: my wife Chao, my parents Norbert and Felizitas, and my sister Miriam. Thanks to my office mates Oliver and Myles as well; it was always fun with you. And finally, I thank my supervisors Novi and David.

CONTENTS

ACRONYMS

AE	autoencoder
CNN	Convolutional Neural Network
DP	demographic parity
EOdds	equalised odds
EOpp	equality of opportunity
ERM	Empirical Risk Minimisation
GAN	Generative Adversarial Network
GP	Gaussian process
INN	Invertible Neural Network
LR	Logistic Regression
ML	machine learning
MLP	multi-layer perceptron
MMD	Maximum Mean Discrepancy
NN	(artificial) neural network
SVM	Support Vector Machine
TNR	true negative rate
TPR	true positive rate
VAE	variational autoencoder
cVAE	conditional VAE

NOMENCLATURE

P	Probability
s	Sensitive attribute/spurious attribute/subgroup label
S	Random variable for the sensitive attribute/spurious attribute/subgroup label
$\mathcal{S}$	Set of possible values for the sensitive attribute/spurious attribute/subgroup label
$\boldsymbol{x}$	Input features (without the s attribute)
y	Class label (ground truth)
Y	Random variable for the class label
$\mathcal{Y}$	Set of possible values for the class label
$\hat{y}$	Predicted label
$\bar{y}$	Fair target label
$\boldsymbol{z}$	Encoding of $\boldsymbol{x}$

GLOSSARY

Adult/Census Income	A popular fairness dataset based on census data from the U.S.
CelebA	Face attributes dataset with more than 200K celebrity images
MNIST	Dataset of handwritten digits
demographic group	Set induced by the sensitive attribute
fairness definition	An aspirational (often legally inspired) specification of a fair classifier
fairness metric	A metric which quantifies how well a fairness definition is satisfied
sensitive attribute	An attribute that, usually for legal or ethical reasons, should not be the basis for classification
spurious attribute	An attribute that is correlated with the prediction target in the training set but not in the deployment setting
subgroup label	A label indicating subgroups that should all be handled equally well by the classifier

Part I

PRELIMINARIES

This part covers the introduction, the related work and a summary of the work presented in part II.

1 | INTRODUCTION

1.1 Problem statement

In order for machine learning (ML) systems to be used more widely, they have to become more trustworthy (HLEG AI, 2019). The susceptibility of deep (artificial) neural networks (NNs) to adversarial attacks has been well documented, but there are other problems as well. This includes their opaqueness and their general tendency to take shortcuts; leading to situations where neural networks do not do 'what we meant', but just what they were explicitly told to do.

This problem becomes especially severe when the training data is biased in some way, which, by default, makes the ML system internalise the bias, or, in some cases, even exacerbate it. Consequently, when applied in the deployment setting, the system will not behave in the desired way. The topic of this thesis is dealing with biased data, where the bias is inextricably linked to a special attribute s.

1.2 Motivation and aims

Many datasets with person-related features display biases when examined by common fairness criteria. This can range from relatively harmless biases, like men being, on average, older than women in the CelebA dataset (Liu et al., 2015), to more serious ones, like black men being several times less likely to receive bail than white men in the COMPAS dataset (Angwin et al., 2016). In the absence of truly 'fair' datasets, our methods have to be able to avoid these biases.

Throughout this thesis, the aim is to learn a model from biased training data, which gives fair predictions (where 'fair' is given by specific definitions) on an unbiased test set. It is thus not sufficient to simply optimise the cross-entropy on the training set; we have to change the optimisation target to achieve our stated goal. While it is possible to extend notions of fairness beyond classification tasks, the vast majority of work in this area concerns classification only and that will be the case here as well.

This thesis will discuss two kinds of dataset bias: *label bias* and *sampling bias*. In both cases, we consider a classification problem in which the class labels y need to be predicted from input features $\boldsymbol{x}$.

In all cases, there is a special attribute s associated with each input. This attribute can have different meanings: In the setting of *label bias*, s usually encodes membership in a demographic group, such as gender, but more generally, it is a feature that should not be used to make predictions for legal or ethical reasons. (See below for other possible meanings of s.) In this setting, we call s the *sensitive attribute* (because it carries sensitive information). We will often refer to the set of all samples which share a specific sensitive attribute as one *demographic group*. In most of the examples, s and y are binary variables, but this does not have to be the case. Crucially, however, the label bias is related to s in a very specific way: depending on the value of s, labels are either flipped from $y = 0$ to $y = 1$ or vice versa, i. e., there is an error in the labels which is correlated with the sensitive attribute. This could, for example, be result of societal discrimination of demographic groups.

For *sampling bias*, there is also a special attribute s, but it does not necessarily have to denote a *sensitive* attribute; it can more generally be a *spurious* variable which is correlated with the class label y in the training set, but is not truly predictive of y in the general case. Alternatively, it might refer to a natural *subgroup* of the y classes. Sampling bias then means that the training set is not uniformly sampled from the underlying distribution: Instead, sampling depends on s and y; e. g., there might be almost no samples of $s = 0$ and $y = 1$ in the training set. The effect is that ML models are tempted to take shortcuts and use s as a shorthand for y.

Figure 1.1 shows a visual representation of the two kinds of dataset bias discussed here.

The overall goal in all cases is to make the classifier invariant to s. We can measure the invariance to s directly by computing *fairness metrics* for the predictions of the classifier. In contrast to the accuracy metric, these fairness metrics give a more complete picture of how invariant the classifier is. This is especially true if evaluation is done on an imbalanced set – for example in the event that a truly unbiased test set is unavailable. In such cases, accuracy can be highly misleading, because good performance on the majority class can hide poor performance on a minority class. Fairness metrics do not suffer from this problem because they specifically look at the results in the different subgroups. However, we typically try to evaluate our models on a test set that is as unbiased as possible, in order for accuracy to be meaningful.

Fairness metrics require a fairness *definition*, and the fairness definition with the clearest interpretation in the described setup of label bias and

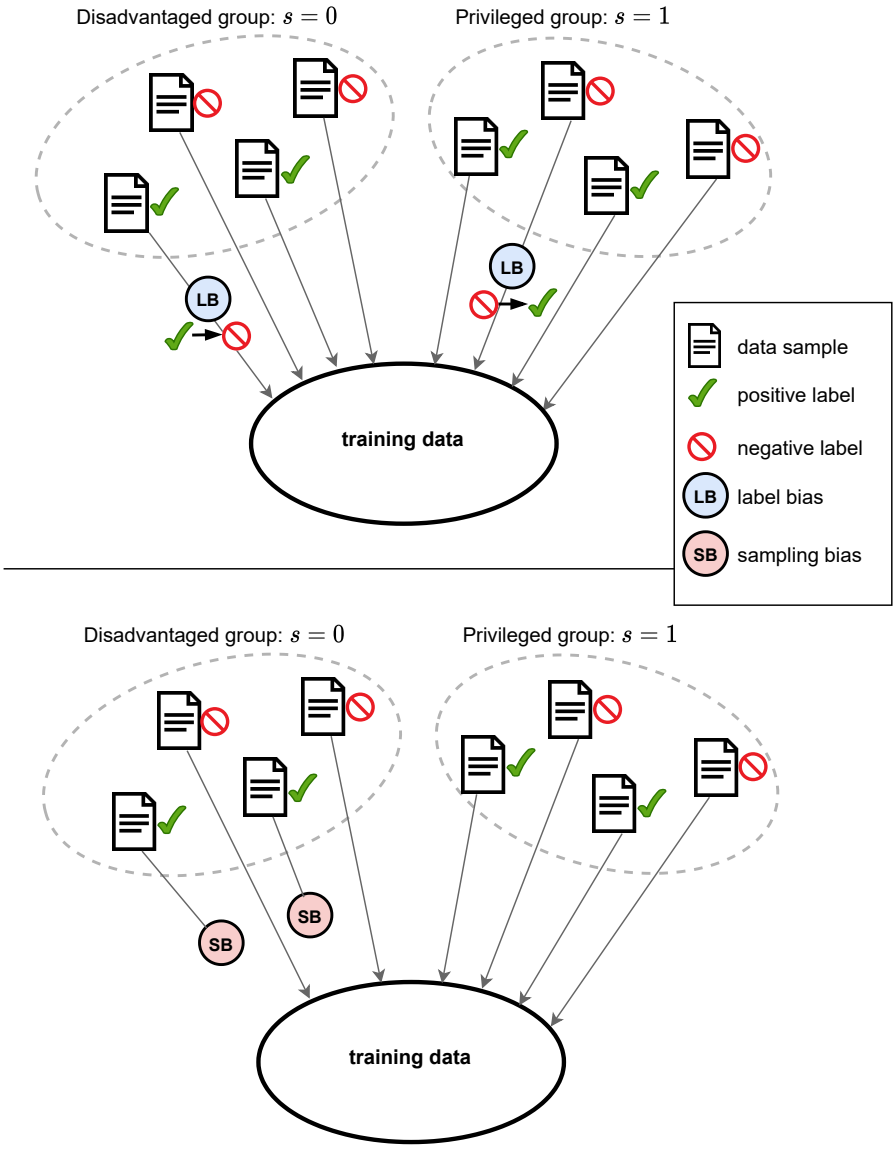


Figure 1.1: Schematic representation of label bias and sampling bias. TOP: the label bias changes positive labels to negative labels for $s = 0$, and in the opposite direction for $s = 1$ (though this not always the case). BOTTOM: the sampling bias *intercepts* samples with a positive label from $s = 0$.

sampling bias is *demographic parity* (DP), also called *statistical parity* or *independence*. It demands that the predictions $\hat{y}$ be independent of the sensitive attribute s. So, for binary s and y:

$$P(\hat{y} = 1|s = 0) = P(\hat{y} = 1|s = 1)\,. \tag{1.1}$$

There are multiple DP *metrics* which track how close the predictions are to satisfying the equality, among them, the difference and the ratio of the terms on the two sides of the equation.

When talking about *un*biased datasets above, we did not specify exactly what this meant, and that is because the definition thereof can differ from task to task, but one way to define it is as a *balanced* dataset where all combinations of s and y occur at the same rate:

$$P(y = 0, s = 0) = P(y = 0, s = 1) = P(y = 1, s = 0) = \ldots \tag{1.2}$$

In such a dataset, we have $y \perp s$, and thus, perfect predictions ($\hat{y} = y$) on this dataset will satisfy $\hat{y} \perp s$ and hence DP. We can conclude that perfect accuracy on a balanced test set implies demographic parity (the reverse does not hold). However, if a model's predictions satisfy DP on a *biased* dataset, then they cannot be perfectly accurate with respect to that dataset's biased labels anymore, which makes sense because the goal is to be accurate to the *unbiased* dataset. This leads to a fairness-accuracy trade-off on biased test sets.

The other two fairness definitions which are commonly used are *equality of opportunity* (EOpp) and *equalised odds* (EOdds), which do not require the prediction $\hat{y}$ to be independent of s, but they do require that the model makes equally high-quality predictions for all values of s. Concretely, for EOpp, the *true positive rates* (TPRs) need to be the same for all demographic groups (again for binary s and y):

$$P(\hat{y} = 1|y = 1, s = 0) = P(\hat{y} = 1|y = 1, s = 1)\,, \tag{1.3}$$

which is also required by EOdds, but EOdds additionally requires the same of the true negative rates (TNRs):

$$P(\hat{y} = y'|y = y', s = 0) = P(\hat{y} = y'|y = y', s = 1) \quad \forall y'\,. \tag{1.4}$$

Just like DP, the fairness definitions EOpp and EOdds can help us evaluate how much a classifier is affected by the bias in the training set.

1.3 Relations to other fields and clarification of terms

This thesis is principally written from the perspective of algorithmic fairness, but it touches on other fields as well, like *domain shift* and *causality*. Furthermore, the thesis makes use of concepts like *transferable representations* and *interpretability*. I summarise the areas here and discuss their relevance to the presented work.

A domain shift is, in general, a change in the data distribution between the training set and the deployment setting; one often talks of a "source domain" and a "target domain". Typically, this leads to poor performance of the ML system in the deployment setting, which necessitates the development of *domain adaptation* methods. There are thus strong parallels to the problem of dataset bias and fairness, but domain shift is more general and has a different emphasis. In domain shift, it is the whole data distribution that changes, including the meaning of individual features, whereas in the biased-data setting (as defined in this thesis), the distribution of training data broadly matches that of the deployment setting, except for incorrect labels or censored sampling. Furthermore, the field of algorithmic fairness is characterised by a focus on special attributes (sensitive attributes) that should not be used to make predictions; a focus that the problem of domain shift lacks. Nevertheless, methods developed for domain adaptation can often be adapted to the problem of biased data: A not insignificant amount of the prior work discussed in chapter 2 was motivated by domain adaptation instead of fairness. Similarly, it should be possible to adapt fairness methods to tackle domain shift, however, this has rarely happened.

A related area is *transfer learning*, which is usually summarised as: learning on one task and transferring the knowledge to a different task. The transfer can happen without any additional training (zero shot), or, more commonly, with fine-tuning on a small (few shot) or large amount of additional data. Transfer learning is not a direct goal of this thesis, but learning *transferable representations* is, which is a much narrower goal. In the context of this thesis, transferable representations are understood to be representations of input features that are (ideally) suitable for *all* tasks that the original data is suitable for, except for predicting the sensitive/spurious attribute s. This is in contrast to representations that are only useful for predicting one specific target.

The other closely related topic is *causality*. The argument that a solution to the dataset bias problem involves causality goes as follows: As mentioned above, the goal is to ignore the bias in the data and learn the true underlying structure that is hidden in the data. Often, the most fundamental structure

that we can learn is the *causal* structure of the problem of interest. For example, if an ML model truly understood what makes someone a good employee, it would not need to rely on surface characteristics like whether the applicant's name sounds foreign. However, the work in this thesis is not presented under the banner of causality, for several reasons. First, in as far as the presented methods solve a causal structure problem, they solve a very limited one; inputs are mapped to outputs, and no detailed exploration of a potential implicit causal model is performed. Second, the discovered robust structures are not necessarily *causal* in nature, in a narrow sense of the word. This is especially true in image data: For example, while discovering that the presence of a smile and the gender of the person in a photograph are two distinct characteristics requires a deep understanding of human faces, it does not require knowing how smiles and gender are related via cause-and-effect in the real world.

In the chapter on related work (chapter 2), I discuss several methods that apply proper causal methods to fairness, but those methods assume that the true causal structure of the problem is already known, and are not helpful for discovering such structures. To work properly, they need access to either Bayesian networks or structural equation models. Neither of which the methods presented in this thesis can provide.

Finally, a note on the terms "interpretability" and "explainability" in the context of ML and artificial intelligence: many different definitions have been proposed (Barredo Arrieta et al., 2020), but no consensus has emerged yet on what these terms should mean. In the context of this thesis, the term "interpretable" is applied to information, and is taken to mean that these pieces of information are provided in a form that is readily understandable and inspectable by humans. In this sense, a visualisation of an embedding or representation as an image in the original data domain is interpretable, but a 100-dimensional vector of floating point number is not.

1.4 Structure of document

In chapter 2, I present related work, concentrating on the area of algorithmic fairness and dataset bias. Chapter 3 summarises the main contributions of this thesis and describes their relationship to one another. The chapter concludes with a list of the publications that constitute the main contribution of the thesis, accompanied by a description of the contributions to the publications, separated by author. Chapters 4–6 reproduce these publications with minimal changes. In the final chapter, chapter 7, I present conclusions from the main work and directions for future work.

2 | RELATED WORK

2.1 Two views of the dataset bias problem

As alluded to in the introduction, there is a divide in the relevant literature in how the problem of dataset bias is approached. On the one hand, there is the "ground-truth-centric" view – expressed through most of the introduction – that the training set is biased but the deployment setting is not. By *deployment setting* I am here referring to the real distribution that will be encountered by the machine learning system once it is in use.

An instance of this setting has been formalised by Blum and Stangl (2020). In their formalisation, the training set starts out unbiased, but then, for samples with $s = 0$, class labels y are flipped from 1 to 0 with probability ν. In addition, samples with $s = 0$ and $y = 0$ are dropped with probability $1-\beta_{NEG}$ and those with $s = 0$ and $y = 1$ are dropped with probability $1-\beta_{POS}$. Meanwhile, the test remains unbiased and is the basis for evaluating our models.

On the other hand, there is the "definition-centric" view that, given some training data and not necessarily knowing anything about the deployment setting, we define *fairness criteria* which a fair classifier should satisfy. Here, the primary goal is not to correct for the deviation of the training data from the true distribution, but simply to ensure that the classifier is fair in some specific sense. By default, a classifier would *not* be fair in that specific way. It is left open as to whether this unfairness originates from the training data or from the algorithm itself. Evaluation is typically performed on a test set that has all the same biases as the training set, and thus gives rise to a fairness-accuracy trade-off.

The first works in the field almost exclusively took the definition-centric view, most of them using demographic parity (DP) (Dwork et al., 2012) as the fairness criterion.

The boundary between these views is not completely clear-cut. Consider the case where a biased training set is available, and an unbiased test set is assumed to exist, but we do not have access to it (Jiang and Nachum, 2020). Then we might at least know that the unbiased test set satisfies DP. In this case, testing the classifier for DP is an indication for how well it would fare on the unbiased test set.

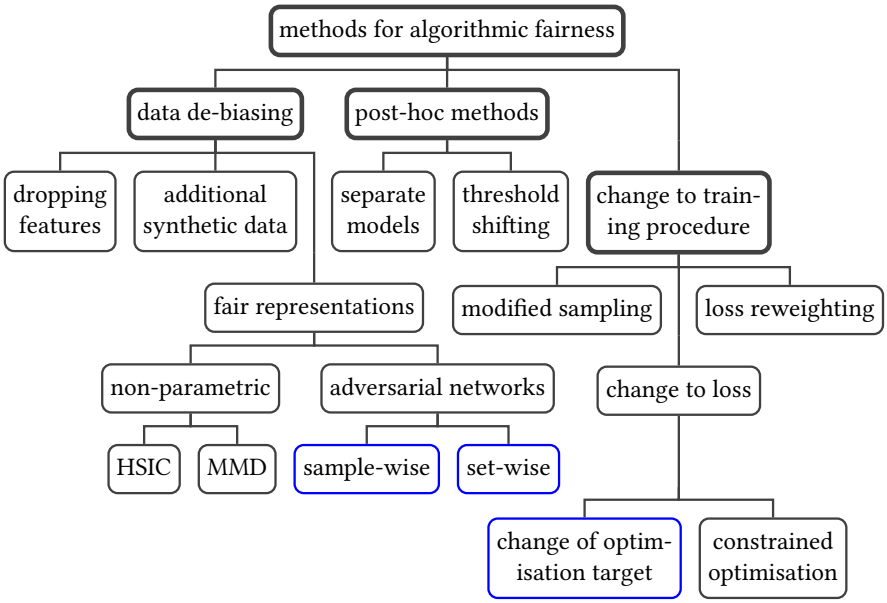


Figure 2.1: Taxonomy of fairness methods. The three categories with blue borders correspond to those categories under which the proposed models in this thesis fall (see chapter 3 for details). This diagram does not show all the axes along which methods can vary. In particular, it does not distinguish between different intended *settings*.

Furthermore, there is significant overlap in the solutions employed for the two views: a method developed for one of them can often be adapted to the other as well. The works discussed in the beginning of this chapter are mostly taking the definition-centric view. Towards the end, I discuss more works with the other viewpoint. Figure 2.1 provides a high-level overview of one possible categorisation of the methods presented here. It can serve as a map for navigating through this chapter.

2.2 Foundations of algorithmic fairness

This section discusses the publications that lay the ground work for the algorithmic fairness literature. More recent publications are discussed in section 2.3.

2.2.1 *Precursors*

The first published work to address the problem of biases in machine learning was arguably the work by Pedreshi et al. (2008) with the main focus being data-mining. Their first fundamental conclusion is that it is not sufficient to remove the protected attribute (which should not be used as a basis for prediction) from the data. That is, if the information about, for example, race is removed

from the data, there is usually enough background information to reconstruct that attribute. In their context of data mining, they distinguish between direct and indirect discrimination: the former explicitly uses discriminatory features in the premise of the rule, while the latter uses features that are closely associated with discriminatory features[1].

2.2.2 *Modification of labels for fairness enforcement*

Independently of Pedreshi et al. (2008), but with a similar intention, Kamiran and Calders (2009) considered discrimination in classification tasks. Their goal is to modify the dataset so that any classifier trained on it will be "fair"; a condition which is measured with a fairness metric on the test set predictions. The test set is drawn from the same distribution as the training set. To demonstrate their method, they use the "German credit dataset" (Dheeru and Karra Taniskidou, 2017) where the task is to classify people as a good or a bad credit risk.

The authors formulate the problem as it was later done in other works on fairness: there is a set of features x, a binary class label y and a special *sensitive attribute* s which determines the demographic group. The sensitive attribute s is assumed to be binary in the derivation, but does not have to be. $s = 0$ refers to the group vulnerable to discrimination and $s = 1$ to all other individuals. It is further assumed that one of the class labels is generally desirable; for example, it might correspond to being accepted for a loan or being given bail. This class label is referred to as the positive label: $y = 1$.

The authors then define a concrete measure of discrimination (where $\hat{y}$ refers to the prediction of a classifier):

$$Disc := P(\hat{y} = 1|s = 1) - P(\hat{y} = 1|s = 0)\,. \tag{2.1}$$

It is zero when the individuals in both groups have the same chance to get a positive prediction. Thus, it measures the violation of DP, where $Disc = 0$ corresponds to satisfying DP.

Their algorithm for de-biasing the dataset comprises two steps. First, a well-calibrated classifier (Naïve Bayes in the paper) is trained on the data *without* the sensitive attribute. This classifier is then used to determine a ranking for how likely a positive label is for a given individual. The highest ranked individuals that have label $y = 0$ and are potentially discriminated

1 This is similar to the legal view of direct and indirect discrimination where direct discrimination is when people with a specific protected characteristic are treated worse, whereas indirect discrimination is a policy that does not make direct references to a protected characteristic but disproportionally affects people with a specific characteristic.

against ($s = 0$) get "promoted" to $y = 1$ and the lowest ranked individuals with label $y = 1$ and $s = 1$ get "demoted" to $y = 0$ until the dataset is fair according to the *Disc* measure. The idea being that the ranking ensures that the changes in the dataset happen to the most appropriate candidates (those closest to the decision boundary); the main goal being a prevention of a big drop in accuracy of the final classifier on the test set whose distribution matches the training set.

This work was extended in Calders et al. (2009), where the same discrimination criterion is used as before. They present a different technique based on giving training examples different weights instead of re-labelling them. Examples with $y = 1$ and $s = 0$ get higher weight than those with $y = 0$ and $s = 0$. For $s = 1$, those with $y = 0$ get higher weight than those with $y = 1$. Weighting means that when the training examples are sampled from the dataset, those with higher weight are used more often. The two approaches are compared on the Adult/Census Income dataset (Kohavi, 1996) where the task is to estimate if someone earns more than $50K per year; the sensitive attribute being *sex* ("gender" in some publications).

2.2.3 *Fair classifiers*

Another common approach is to put fairness *constraints* on the classifier instead of manipulating the dataset. Calders and Verwer (2010) proposes three such approaches, which potentially give more control over the prediction bias than changing the datasets. All three approaches are based on Naïve Bayes and again aim to enforce DP in the predictions, as defined before. The first approach shifts the predicted probabilities in a post-processing step after training the classifier normally. To this end, the sensitive attribute is treated differently than the other features in the Naïve Bayes model. Instead of the class being the cause of the sensitive attribute, the sensitive attribute is regarded as one cause for the class. This is a significant change in perspective that enables a better intuition about the problem. In this modified Naïve Bayes, the conditional probability $P(y|s)$ is then modified in the already trained model until the discrimination score is below a given threshold. Care is taken to not distort the predicted distribution too much with respect to the unfair model. The end result is a model that is (ideally) unbiased but still relies on the sensitive attribute for predictions. This is unworkable if the sensitive attributes are not always available when making predictions. Nevertheless, the algorithm can be considered an improvement over manipulating the dataset, as the resulting bias can be controlled to a much finer degree.

The second method in Calders and Verwer (2010) is based on training two separate models for each sensitive attribute. That is, the dataset is split in two where one half contains all examples with $s = 0$, we call this dataset D_0, and the other half contains all examples with $s = 1$, called D_1. For prediction, we choose either the model that was trained on D_0 or the one for D_1, depending on the sensitive attribute of the example. This method relies on the availability of the sensitive attribute for prediction, like the previous method. The two models are separately tweaked to produce overall fair results. In order to minimise the effect on the accuracy, the models are tweaked the same amount each, but in opposite directions. Conceptually, having two different Naïve Bayes model depending on the (binary) sensitive attribute is equivalent to using one Naïve Bayes model in which the sensitive attribute is connected to all other features. This means that the bias is in all of the features and not just in the class label. The latter was the assumption of the previously discussed model. The difference between these assumptions is not explored in much detail in the paper.

The third method introduces a new latent variable, called L in the paper. L is intended to be an unbiased target class label that replaces the biased class label y from the training data. L can only be estimated and is unbiased in the sense that it is statistically independent from the sensitive attribute: $P(L|s = 0) = P(L|s = 1)$. The observed class label y then only depends on the latent label and the sensitive attribute. L is found by using expectation maximisation to iteratively search for a value that maximises the likelihood of the dataset. The search is restricted by enforcing that L must be equal to y except when $s = 0$ and $y = 0$ or when $s = 1$ and $y = 1$ because these are the two cases where we expect discrimination. There is other prior knowledge that can be incorporated into the search.

The authors present experimental results for the three methods on an artificial dataset and the Adult/Census Income data that Calders et al. (2009) used before. The artificial data has biased and also unbiased class labels, the latter of which are generally not available for real-world datasets, and the data conforms to the assumptions made for the third model. (The existence of the unbiased ground-truth labels makes this paper an example of the ground-truth-centric view.) On both datasets, the two simpler methods outperform the third method that is based on fair latent class labels; the two first methods perform approximately equally well.

Kamishima et al. (2012) take a closer look at the ways in which training data can be biased (or more generally of poor quality) and condense it to three main causes: *prejudice*, *underestimation* and *negative legacy*.

1. *Prejudice*: a statistical dependency between the sensitive attribute and the class label or the (non-sensitive) input features. If the dependency is with the class label, then the paper calls it *indirect prejudice*. If it is with the non-sensitive features, they call it *latent prejudice*. In line with Pedreshi et al. (2008), *direct prejudice* refers to classifiers that make direct use of the sensitive attribute. Kamishima et al. (2012) do not consider latent prejudice to be immediately harmful, but it can be a problem with respect to the protection of personal data and compliance with laws.

2. *Underestimation*: non-convergence of machine learning algorithm due to limited amount of data. This magnitude of this problem can be estimated by considering the difference between the actual training sample distribution and the distribution that the machine learning model has internalised. This is the underlying cause of models that make predictions that are even more unfair than the dataset.

3. *Negative Legacy*: sampling bias and wrong labels in the dataset. In contrast to the problem of *prejudice*, *negative legacy* might not be detectable by analysing the dataset. *Sampling bias*, meaning that certain data points simply are missing, and *wrong labels* can only be corrected if other sources of information are available.

We see that previous works nearly exclusively dealt with the first of these issues, *prejudice*, and more precisely *indirect prejudice*. Kamishima et al. (2012) also focus on *indirect prejudice*; mostly because it is the easiest to deal with, apart from *direct prejudice*. They use Logistic Regression (LR) as the basis for their fair classifier. Their approach to enforcing fairness is qualitatively different from the previous proposals. Instead of manipulating the dataset or using explicit algorithm to make a classifier fair, Kamishima et al. (2012) treat the fairness constraint like a regulariser. A term is added to the objective function that estimates the mutual information between y and s and gets minimised together with the other parts of the objective function. A factor in front of the regularisation term determines how much to value fairness over accuracy.

2.2.4 *Fairness based on similarity*

Nearly all previously discussed papers express some dissatisfaction with the demographic parity (DP) fairness definition. Thanh et al. (2011) use a very different fairness metric based on the idea that similar people should be treated similarly. The setup is still that the training set is biased, only a

different fairness definition is used to judge classifiers. The authors first define a distance metric which defines a neighbourhood, based on the Manhattan distance on the z-scores of the attributes. The distance metric is supposed to be only making use of legally admissible attributes. An individual is then considered to be unfairly treated if it is classified differently than its neighbours. More concretely, for any data point we can check how many of the k nearest neighbours have the same class label as that data point. If the percentage is under a certain threshold then – so the authors argue – there was discrimination against the individual corresponding to that data point. In order to create a fair classifier, for this definition of fairness, the authors propose to pre-process the dataset, similar to Kamiran and Calders (2009) before, by flipping the class labels of those data points where the class label is considered wrong or biased. The experiments in the paper show that this method is successful in reducing the discrimination (according to the given criterion) for a range of classifiers on the Adult/Census Income dataset.

Dwork et al. (2012) give a more in-depth analysis of the idea of distance-based fairness and DP. They present explicit detailed criticism of the DP metric, mainly in the form of several scenarios in which DP is maintained, but the system treats a lot of individuals very unfairly. In one scenario, the sensitive attribute carries important information and removing it makes everyone worse off. Another scenario considers the case of *subset targeting*, where there is discrimination against subgroups which are not covered by the DP metric that only considers the major demographic groups. (This problem has also been referred to as *hidden stratification*.) Defining fairness via group membership will always leave open the possibility of discriminating against ever smaller subgroups down to the level of individuals.

Based on these considerations, Dwork et al. (2012) argue for *individual fairness* instead of *group fairness*. They propose a method based on a similarity metric, but do not give a general recipe to construct such a metric; instead stating it has to be constructed on a case-by-case basis.

The proposed method is then as follows: a classifier is fair *if and only if* the predictive distributions for any two data points are at least as similar as the two points themselves, according to a given similarity measure for distributions and a given similarity measure for data points. They call this condition the *Lipschitz condition*. The authors propose two practical similarity measures for distributions that are well-known in the respective literature. In order to train a fair classifier, the Lipschitz condition is then used as a constraint for the optimisation.

2.2.5 *Fair representations*

Zemel et al. (2013) approach the problem of biased training data more directly by seeking to transform the features, such that all traces of the sensitive attribute are removed, while at the same time letting the features still carry the information required for predicting the class label. This is different from Kamiran and Calders (2009) where the dataset was transformed as well, but there, the change was to the *labels*. The idea of transforming the features assumes that the main problem with the data are unwanted dependencies between s and the other features (called *latent prejudice* in Kamishima et al., 2012), and that the training labels are mostly unproblematic. The intended result is then that classifiers trained on the transformed data will make s-invariant predictions 'by default', because they do not know about s.

More specifically, the stated goal of Zemel et al. (2013) is to achieve both *group fairness* (in the sense of Demographic Parity) and *individual fairness* (in the sense of treating similar people similarly), as Dwork et al. (2012) did before. Let z denote the learned fair representation. The condition for fairness is then

$$P(s = s'|z = z') = P(s = s') \quad \forall s' \in \{0, 1\}, z' \in \mathbb{Z} \,. \tag{2.2}$$

Zemel et al. (2013) propose to map the biased inputs x to *prototypes* in the same space as x. The probability for being assigned to a particular prototype must be the same for inputs with $s = 0$ and for $s = 1$. Based on a given distance function (or similarity measure), inputs are more likely to be assigned to prototypes that are close-by. The classification task is then based entirely on the prototypes (the fair representation z). In their work, a linear model is used to map the prototypes to the outcomes y. Aside from the distance function, the whole model has therefore only two kinds of parameters: the locations of the prototypes and the weights in the linear model. These parameters are all optimised together. The objective function consists of three terms: the first enforcing demographic parity (DP) via the prototype locations, the second ensuring that the prototypes are close to the inputs and the third trying to maximise accuracy by adapting the weights of the classifier. The experiments are done on the German credit dataset, the Adult / Census Income dataset and a dataset based on the Heritage Health Prize milestone 1 challenge (ForeverData.org, 2015) where the goal is to predict how many days a given person will spend in the hospital in a year.

Feldman et al. (2015) try to improve upon the work by Zemel et al. (2013), focusing on fair representation as well. The authors try to ground their definition of fairness in U.S. law. They define *Disparate Impact* (DI) as

$$DI = \frac{P(y = 1|s = 0)}{P(y = 1|s = 1)} . \tag{2.3}$$

With a target of $DI = 1$, this would simply enforce DP. However, based on some rulings and recommendations of the U.S. legal system, they advocate the 80% Rule, which states that DI should not be below 80% (or above 125%). This means that the acceptance rate of a disadvantaged demographic group should not be less than 80% of the acceptance rate of the other group. With this definition, there is an explicit allowed range for small unfairness. Previously, researchers just tried to get as close as possible to $DI = 1$, without stating how close is close enough.

For measuring the fairness of the (non-sensitive) input features (disregarding class labels for the moment), the authors define ε-fairness. The features x are ε-fair, if any predictor that tries to predict s from x can only achieve a balanced error rate that is higher than ε. Feldman et al. (2015) prove that ε-fairness with a suitable ε is incompatible with violating the 80% rule. That is, if the sensitive attribute cannot be predicted from the features, then a classifier trained on that data will automatically be fair (to a certain degree). For datasets that are extremely unbalanced, the required ε approaches 1/2 which corresponds to absolutely no information in x about s. Note that the definitions require that the performance of the best possible classifiers is known, which is rarely the case. In the paper, a Support Vector Machine (SVM) classifier is used to measure the ε for ε-fairness.

In order to create a fair dataset, the authors present an algorithm that considers every feature individually and shifts the values such that the distributions $P(z = z'|s = 0)$ and $P(z = z'|s = 1)$ are identical (z refers to the shifted values, x refers to the original values). This shift retains the ordering of the data points with regard to that feature. This is to ensure that z can still be used to predict y (which assumes that y and s are sufficiently uncorrelated, so that z can at the same time be unpredictive of s and predictive of y). The method only works on numerical features. The paper contains two other algorithms for removing unfairness which are not as invasive, meaning some amount of unfairness remains, but the ability to predict is improved. They are referred to as *Partial Repair* algorithms, as opposed to full repair. This is the fairness-accuracy trade-off that basically all works in this area consider. Both of the other methods try to preserve the ranking of the data points

with respect to the individual features while minimising the distance of the distributions for the different groups.

This work by Feldman et al. (2015) is a step up from the work by Zemel et al. (2013) because the representation (ideally) is fair with regards to any machine learning algorithm, not just one particular. Furthermore, theoretical bounds were proved for the expected bias of a classifier. However, the work does not take into account individual fairness in any way.

While the algorithms by Feldman et al. (2015) are manually constructed and explicit, Louizos et al. (2016) use an approach that falls more in the area of end-to-end learning where fewer hand-crafted algorithms are used. The method is based on deep variational autoencoders (VAEs), with an encoder and a decoder that are both modelled as deep neural networks. The encoder produces the distribution of the latent (fair) representation z from the original features x and the sensitive attribute s. The decoder recovers the distribution of x from z and s. By choosing a factorised prior $P(s)P(z)$, a separation between s and z is encouraged.

This method can be improved further by taking into account the labels when constructing the fair representation. If this is not done, z loses the ranking information from x (see Feldman et al. (2015) above). To this end, a second latent variable is introduced, $\tilde{z}$, which encodes the variation in z that is not explained by the class labels y. z is then determined by y and $\tilde{z}$, and x is determined by z and s as before. $\tilde{z}$ and s have independent priors. This structure ensures that y can be predicted from z. However, this introduces a new problem: if y is correlated with s, then z will be as well. To overcome this, an additional penalty term is introduced that forces $P(z|s = 0)$ and $P(z|s = 1)$ to be as close as possible. This is realised with a measure of distance between distributions called Maximum Mean Discrepancy (MMD).

To test whether s can be recovered from z, a Random Forest model and a Logistic Regression model were trained to predict s from z. Using MMD for an additional unfairness penalty, seems to improve fairness. When training a classifier on z to predict y, there is a small drop in accuracy.

2.2.6 *Other fairness criteria*

The early literature on fairness in machine learning used only DP and similarity-based fairness to measure the bias in predictions. Kleinberg et al. (2016) formalised three different group fairness conditions. The authors consider a scenario where the data points x are sorted into bins b and each bin is associated with a prediction score $f = P(\hat{y} = 1|b)$ where $\hat{y}$ is the predicted class label and x the (non-sensitive) features.

1. Calibration within groups: If the prediction score for a given bin is f, then when considering all the data points with group s in the bin, a fraction of f of those should have the class label $y = 1$. In other words, the prediction score f is well calibrated with respect to group s. This should be the case for all groups.

2. Balance for the negative class: The true negative rate should be the same for both groups: $P(\hat{y} = 0|y = 0, s = 0) = P(\hat{y} = 0|y = 0, s = 1)$.

3. Balance for the positive class: The true positive rate should be the same for both groups: $P(\hat{y} = 1|y = 1, s = 0) = P(\hat{y} = 1|y = 1, s = 1)$.

The first criterion essentially ensures that the predictor works correctly for both groups. This prevents a classifier from predicting one group correctly but always returning a negative answer for the other group. The second and third put emphasis on negative and positive class labels respectively. In criterion 2, we allow the classifier to mis-classify those data points with $y = 0$, but we want to make sure that the misclassification rate is the same for both groups. In other words, members of the different groups have the same chance to get a correct classification if they should receive a negative classification. The same holds for criterion 3 and positive classifications ($y = 1$).

A plausible question is whether it is possible to achieve all of these criteria simultaneously. The authors show that a perfect predictor, i. e.one that gives a score of $P(\hat{y} = 1|x) = 1$ to data points with $y = 1$ and $P(\hat{y} = 0|x) = 1$ to those with $y = 0$, automatically satisfies all three criteria. (The same is in general not true for DP: if the test set labels exhibit a statistical dependency to s, then a perfect predictor does the same.) Furthermore, the authors show that if the dataset is fair in the sense that it satisfies $P(y = 1|s = 0) = P(y = 1|s = 1)$, i. e., the base acceptance rate is the same for both classes, then a "random" classifier which assigns that base rate as the prediction score to all data points indiscriminately also satisfies all three criteria. For this case, DP is satisfied as well. The paper provides a proof that those two cases are the only ones that achieve the three presented guarantees simultaneously. Demographic Parity can only be achieved simultaneously with these in the second scenario ("random classifier"). The general case sits between those extremes: the predictor is not perfect and the test set is not unbiased, and therefore, these definitions of fairness are not compatible. Note, however, that criterion 2 and 3 *are* compatible with one another.

The paper's main contribution is a theorem showing that the three criteria are in general incompatible, even if we only consider approximations of them.

The conclusion to draw is that it is remains difficult to choose the appropriate definition of fairness to judge a classifier.

Hardt et al. (2016) expand on criterion 3 in Kleinberg et al. (2016) and develop a method to enforce it via post-processing. Furthermore, an alternative criterion is introduced that is the combination of criterion 2 and 3. This is also the work that gave criterion 3 the name *equality of opportunity* and that popularised the name *demographic parity*. The treatment of fairness by the authors is explicitly "oblivious", which here means that only the general statistics are known about the features x, the sensitive attributes and the class labels y. In particular, there is not enough information available to develop a similarity measure which could be used to target *individual fairness*.

In this "oblivious" setting, the authors define their fairness criteria in terms of statistical independence: equalised odds (EOdds) refers to the case where $\hat{y}$ and s are independent conditional on y. This is equivalent to the true positive rate and the false positive rate being the same for all groups. Equality of opportunity (EOpp) is, as mentioned above, the case where just the true positive rate is the same for all groups. This means that $\hat{y}$ and s are independent conditional on $y = 1$. One of the main advantages of these definitions compared to DP is that a perfect predictor satisfies them on any evaluation set.

In order to construct a fair classifier out of a *binary predictor* (giving only hard binary predictions $\{0, 1\}$) via post-processing, the output is randomised in such a way as to remove the bias. If the overall probability for a positive prediction of the unfair classifier is given by $P(\hat{y} = 1|s)$, then the randomised probability for the fair positive predictions $\tilde{y} = 1$ is given by:

$$P(\tilde{y} = 1|x, s) = \sum_{y' \in \{0,1\}} P(\tilde{y} = 1|\hat{y} = y', s) P(\hat{y} = y'|x, s) \tag{2.4}$$

There are 4 free parameters $P(\tilde{y} = 1|\hat{y} = y', s = s')$ with $y' \in \{0, 1\}$ and $s' \in \{0, 1\}$. As an example, if the unfair predictor predicts $\hat{y} = 0$ for an input with $s = 0$, then the fair predictor predicts $\tilde{y} = 1$ with probability $P(\tilde{y} = 1|\hat{y} = 0, s = 0)$ which might be non-zero. In addition to the fairness condition, we also want to enforce accuracy. To that end, the authors introduce a loss function $\ell(\tilde{y}, y)$ that quantifies the cost of predicting the wrong class label. The final optimisation problem for the 4 free parameters is then to minimise ℓ under the constraint of EOpp or EOdds and the constraint that the parameters must be valid probabilities.

For a predictor that outputs a *score function*, the post-processing step consists of choosing differing thresholds for $s = 0$ and $s = 1$, such that the predictions become fair. If f is the score, then for a given threshold t we

predict $\hat{y} = 1$ if $f > t$. The two thresholds (one for each group) are found by minimising ℓ with the fairness constraints, as before. To satisfy the constraint, it might be necessary to randomise the result. This is done by using two constraints per group: if f is above both, the result is $\hat{y} = 1$, if it is below both, $\hat{y} = 0$ and if f is between the thresholds, then the result is chosen at random. This method – as well as the one before – requires the sensitive attribute for all predictions at test time, as it has to choose the right threshold.

The authors prove that with this post-processing, the Bayes optimal (but biased) classifier becomes the Bayes optimal unbiased classifier.

2.2.7 *Balancing datasets with synthetic data*

Another way to view the fairness problem is to regard the dataset as missing certain kinds of samples. This falls under the ground-truth-centric view: the *true* data distribution is fair, but our training set only covers a part of it, which means it looks unbalanced or unfair. An idea to deal with this, is to *generate* the missing data, typically with a Generative Adversarial Network (GAN).

Sattigeri et al. (2019) is one such approach. The method is based on a conditional GAN, that is conditioned on the sensitive attribute s. The GAN then produces fake input samples x_F together with corresponding labels y_F. In addition to the usual discriminator which tries to distinguish fake x_F from real x samples, there is another adversary that tries to predict s from y_F, which is opposed by the generator. This is meant to ensure that the labels of the generated samples have the same distribution for all values of s, i. e., that the generated data distribution is balanced in terms of s and y. There is also a discriminator that takes both x_F and y_F as inputs, and determines whether this is a plausible pair. Once this GAN has been trained, a classifier is trained on the generated, balanced data. The expected advantage over other methods of balancing the dataset is that this method produces a more realistic distribution, because the adversarial training ensures that the data "looks real". However, this reliance on the discriminator also means that the generated data cannot contain data from unobserved parts of the data distribution, because such data would very easily be spotted as fake. For example, if a face image dataset does not contain men with lipstick, then the GAN will also not produce such samples (even though it knows about men and it knows about lipstick). Generally, GANs cannot produce something out of nothing, so if the data is missing certain sectors entirely, it is very hard to produce samples from this sector that are *realistic*. This problem can potentially be ameliorated by pretraining the GAN on diverse, unlabelled data.

2.3 Recent developments in algorithmic fairness

This section discusses more recent publications from the fairness literature.

2.3.1 *Fair classifiers continued*

Following the initial publications, several works refined the ideas that had been presented there. A very influential work by Zafar et al. (2017b) presents a more efficient way to train a fair classifier. The authors explicitly discuss the tension between the two goals of making fair predictions and not making use of the sensitive attributes at test time. Using sensitive attributes to make predictions is referred to as *Disparate Treatment* by the authors. Algorithms like the one by Calders et al. (2009) use the sensitive attribute during prediction to achieve a very high degree of fairness, but this can have a lot of problems from a legal perspective and can also easily go wrong. (Making use of s comes close to *direct discrimination* discussed above.) Thus, Zafar et al. (2017b) set themselves the goal of achieving fairness while making use of the sensitive attributes as little as possible, i.e. avoiding Disparate Treatment.

In order to train a fair classifier, Zafar et al. (2017b) use a proxy of the definition of DP that is easier to optimise during training, namely the covariance between the sensitive attribute and the predicted score f:

$$Cov(s, f) = \mathbb{E}[(s - \bar{s})f] - \mathbb{E}[(s - \bar{s})]\bar{f} \approx \frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s}) \cdot f_i \tag{2.5}$$

The authors refer to this as the covariance criterion. The classifier can then be trained in two ways. The first way is to minimise the loss that enforces accuracy with the constraint that the covariance criterion is below a certain threshold. The other way is to minimise the covariance criterion with the constraint that the accuracy is above a certain threshold. The paper includes experiments with these techniques with SVM and Logistic Regression (LR) classifiers on the Adult/Census Income dataset and the Bank marketing dataset (Dheeru and Karra Taniskidou, 2017). In the Bank marketing dataset, the task is to predict whether a client will respond to marketing for a term deposit, based on 20 attributes of the person. The sensitive attribute is *age*. The proposed algorithm performs similarly to the best previous algorithms (similar in fairness and accuracy), notably without making use of the sensitive attribute for predictions.

In a follow-up work, Zafar et al. (2017a) adapt their work to avoid "disparate mistreatment". Avoiding disparate mistreatment is the same as enforcing equalised odds (EOdds), as defined in the contemporary work by Hardt et al.

(2016). As in the previous work, they define a tractable proxy for this measure. To this end, they first define a function g that measures how well the class label $y \in \{0, 1\}$ and the predicted score for a data point $0 \leq f(x) \leq 1$ agree:

$$g(y, x) = \min\left(0, \left(y - \frac{1}{2}\right) \cdot \left(f(x) - \frac{1}{2}\right)\right) \tag{2.6}$$

Subtracting $\frac{1}{2}$ is necessary to centre the values around 0. The proxy for EOdds is then the covariance between the sensitive attribute and g. g can also be modified to only take into account cases with $y = 1$ which corresponds to enforcing equality of opportunity (EOpp). The paper contains experiments on the ProPublica/COMPAS dataset (Angwin et al., 2016). The task for this dataset is to predict whether a criminal offender committed another misconduct or felony within two years, based on personal information that includes age, gender, race and past criminal history.

Woodworth et al. (2017) also consider EOdds as a fairness definition but give strong criticism for the post-processing approach from Hardt et al. (2016). They present examples where finding the optimal (but discriminatory) predictor in a particular hypothesis class and then correcting it post-hoc performs poorly. Certain training distributions allow the training of good fair classifiers from scratch but when doing post-processing on a Bayes-optimal predictor for this distribution, the result can be a very bad predictor that gives essentially random predictions. The authors explicitly construct one such distribution: it corresponds to the case where the sensitive attribute s is more predictive of the class label y than the features x, which becomes an especially severe problem when s is only predictive during training time, but not at test time. See the discussion of *Coloured MNIST* below for such an example.

This topic has since received more contributions than can be listed here. The following is a very short sample. In the direction of Zafar et al. (2017b), there is for example Quadrianto and Sharmanska (2017), Ustun et al. (2019) and Lohaus et al. (2020). In the direction of Kamiran and Calders (2012), there is for example Agarwal et al. (2018) and Roh et al. (2021). In the direction of Hardt et al. (2016), there is for example Hébert-Johnson et al. (2018).

2.3.2 *Fair representations continued*

With the emergence of adversarial neural networks, interest was renewed in creating fair representations. Ganin et al. (2016) can be seen as a direct precursor: they propose adversarial learning for domain adaptation. Their method involves learning a shared representation that is invariant to the

different domains. If we consider demographic groups as different domains, then this is essentially fair representation learning.

This is the premise of Edwards and Storkey (2016). Four neural networks are trained in an adversarial setting: the encoder $f(x)$ which encodes the unfair representation x into a fair representation z, the classifier $g(z)$ which tries to predict y from z, the decoder $k(z)$ which tries to reconstruct x from z, and finally the adversary $h(z)$ which tries to predict s from z. The classifier g ensures that z contains enough information to predict y. The decoder k ensures that the fair representation contains enough information from x and can be used by other classifiers as well. The sign of the adversary's loss is inverted for the gradient of the encoder f, so that the encoder tries to make the adversary's predictions worse. Encoder and adversary should converge to a point where the adversary can only predict the correct s from z at chance level. Assuming that the adversary is powerful enough to detect any trace of s in z, this means that no information about the sensitive attribute s remains in the fair representation z. Note that there is a tension here between requiring x to be reconstructible from z and requiring z to contain no information about s, since x *does* contain information about s.

Beutel et al. (2017) explicitly connect the method presented in Edwards and Storkey (2016) to various fairness definitions.

Zhang et al. (2018) propose an architecture similar to Edwards and Storkey (2016), but with the difference that no decoder (and no reconstruction loss) is used and that the adversary tries to predict s from the final output of the classifier instead of an intermediate representation. The goal is not to learn a fair representation, but a fair classifier.

Madras et al. (2018) is another, similar approach. One difference to Edwards and Storkey (2016) is that several fairness criteria are considered: Demographic Parity, Equality of Opportunity and Equalised Odds are all converted to adversarial objective functions. Another difference is that the decoder k has access to s for the decoding: $k : (Z, S) \to X$; this ensures that there is no tension between reconstructing x and removing s from z. However, the key contribution of the paper is the objective functions for the adversary. In the case of Demographic Parity, we take – for each sensitive group separately – the average absolute difference between what the adversary predicted $h(z)$ and the true sensitive attribute s. After computing these two separate averages, the averages are added and get a negative sign to form the objective function. For Equalised Odds, the average absolute difference is calculated on each sensitive group-label combination (s, y) for $s \in \{0, 1\}$ and $y \in \{0, 1\}$ separately. Computing these separate averages is referred to as *group normalising* in the paper. The authors present a proof for upper bounds on

unfairness by the optimal adversary when these objective functions are used. An alternative objective function could be based on cross-entropy – as used in Edwards and Storkey (2016) – but the authors dismiss this on the grounds that there are situations in which it leads to wrong results. However, in the experiments they show that the proposed objective and one objective based on cross-entropy lead to very similar results.

Based on the weights of the different parts of the loss function (adversarial loss, classification loss, decoder loss, etc.), different trade-offs can be achieved.

2.4 Fairness via causal reasoning

In this section, I will briefly discuss a view that also falls under the definition-centric view: in it, the goal is to make decisions that conform to causality-based notions of fairness. Methods in this category assume that the causal structure of the task is known, which can then be used to identify unfair path ways for making decisions. One motivating example is this (DeDeo, 2014): In college admissions, we might not want to discriminate against prospective students from poorer backgrounds. If the goal is to admit those students that have a chance of graduating, then we could employ a machine learning algorithm to predict graduation rate. One attribute that might turn out to be predictive of graduation is physical fitness. If this attribute's causal influence is through, for example, signalling a character trait such as grit, then this is an admissible attribute. However, if on the other hand physical fitness is caused by access to expensive gyms, then it is a signal for high socioeconomic status which we do not want to use as a criterion. So, depending on what the causal mechanism is, physical fitness could be a discriminatory or a non-discriminatory feature.

Kilbertus et al. (2017) is one of the first to use causal graphs to define fair decisions. In their work, the prediction output $\hat{y}$ is a part of the causal graph that models the data. A simple causal fairness definition would be to disallow any causal paths between s and $\hat{y}$. However, instead of considering paths that start from s directly, the authors argue that the observed proxies for s are more important. A proxy for s is a clearly defined observable quantity that is significantly correlated with s. According to the authors, the sensitive attribute in its pure conceptual form may influence the prediction directly, but any observable proxy of it may not. The given reason is that if all influence of s on $\hat{y}$ was removed, only very few features would remain to make a decision because usually nearly all features are influenced by s in some way.

The concept of removing causal influence can be efficiently expressed with *interventions* on the causal graph. An intervention on a variable v cuts

off all parents of v and sets v to a specific value, say v'. This is written as $do(v = v')$. So, for a given proxy p ($p \in \{0, 1\}$), predictions $\hat{y}$ exhibit no *proxy discrimination* if

$$P(\hat{y}|\text{do}(p = 0)) = P(\hat{y}|\text{do}(p = 1)) \,. \tag{2.7}$$

If this is fulfilled, there is no influence from s that is mediated through p. There might still be direct influence of s on $\hat{y}$. The fact that *do*-interventions are used in the definition means that a causal model is required to evaluate this criterion; observational data does not suffice.

The paper provides an algorithm that constructs a fair classifier (for the provided fairness definition) given a structural causal model (Pearl, 2009). The resulting classifier is fair *by construction*. However, note that this procedure does not lead to true individual fairness, because the influence of p is only removed at the population level (except in the case where all descendants of p are completely removed). Individual decisions can still be unfair as long as it balances out overall.

The authors show a number of properties of their method with corresponding proofs. (Though no experiments are included in the paper.) For example, if the influence of p on x is additive and linear, any predictor of the form

$$g(x - \mathbb{E}[x|do(p = p')])$$

has no proxy discrimination.

A contemporary paper by Kusner et al. (2017) approaches the problem differently. In this paper, a new fairness criterion which they call "counterfactual fairness" is defined making use of causal reasoning. For the definition, they assume access to a causal model that can be used to compute *counterfactuals* (for example a structural equation model; Kaplan, 2008). This is in contrast to Kilbertus et al. (2017) who relied on just *interventions*. (According to Pearl's causal hierarchy (Pearl, 2019), counterfactuals require a deeper level of causal information than interventions.) The causal model depends on unobserved background variables U, non-sensitive features x and a sensitive attribute s. A predictor $\hat{y}$ is then *counterfactually fair* if the following equality of *counterfactual probabilities* holds:

$$P(\hat{y}_{s=i}(U) = 1|x, s = i) = P(\hat{y}_{s=j}(U) = 1|x, s = i) \tag{2.8}$$

where $i, j \in \{0, 1\}$ and $i \neq j$. The first counterfactual probability is actually not a real counterfactual: it is just the probability of observing $\hat{y} = 1$ given the background variables U, the features x and $s = i$. The second counterfactual probability is the probability of observing $\hat{y} = 1$ given U and x and given that

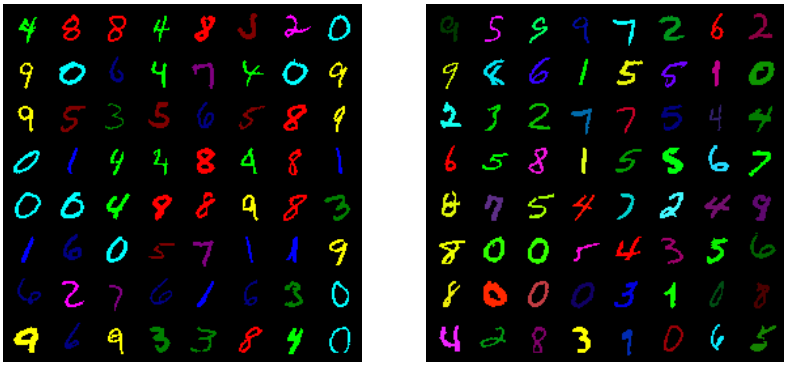
we had $s = j$ instead of the actual value $s = i$. In other words, the criterion demands this: the prediction would have been the same in the counterfactual world where the sensitive attribute is j instead of i. In the counterfactual world, everything that is not causally dependent on s is held constant.

Making sure predictions are the same, for the closest world with $s = j$, is arguably what Dwork et al. (2012) (see above) did with their idea of treating similar people similarly (regardless of s). In this way, counterfactual fairness is similar to the individual fairness that Dwork et al. (2012) defined. However, instead of using a similarity metric, a causal model is needed for counterfactual fairness. The two criteria share some strengths and weaknesses. A strength is that they are both fair on the individual level, which is not the case for Demographic Parity and Equality of Opportunity. In Demographic Parity and Equality of Opportunity, we can achieve fairness by doing "negative" discrimination on some individuals and "positive" discrimination on others, as long as those two effects cancel out. This is because those criteria are defined on *groups* and not individuals. However, in turn this implies that individual fairness and counterfactual fairness could not be used to implement affirmative action ("positive" discrimination for a group) and they rely on the features x being correct.

The authors provide an algorithm to construct a classifier that satisfies counterfactual fairness. However, constructing the causal model remains the major weakness of this approach, and is currently only really feasible for tabular data. As experiments, the algorithm is applied to the Law School Success dataset (Wightman, 1998). The task is to predict the first year average grade (FYA) in law school given the GPA before law school and the score on the entrance exam (LSAT). Also known are race and sex, which are not supposed to be used to make predictions.

If the given causal model is taken as the true model, then it follows that the trained fair classifiers are fair by construction. Additionally, two (unfair) baseline models were trained: one trained on only x and the other trained on x and s; those baseline models indeed turn out to be not counterfactually fair.

Chiappa (2019) have a similar approach, but their work by can be seen as a refinement of Kilbertus et al. (2017). Instead of disallowing influence of *nodes*, this work disallows certain *paths* to influence the outcome. The path pointing directly from the sensitive attribute to the outcome is not allowed, but paths via certain variables are allowed. An example for such a permissible variable comes from the Berkeley admissions dataset (Bickel et al., 1975): overall, the data shows that women were admitted at lower rates, but this turned out to be mediated by department choice. Women were applying to more competitive departments and thus had lower acceptance rates. The



(a) Training set: one-to-one mapping between colour and digit class.

(b) Test set: random assignment of colours.

Figure 2.2: A typical example of the Coloured MNIST dataset.

choice of department C is here an admissible variable: the outcome Y may depend on it. So, a path from S (gender) to Y via C is permissible, and our fairness definition should reflect this. Counterfactual fairness (Kusner et al., 2017) would in this case arguably give the wrong answer. In certain simple cases, path-specific fairness is equivalent to the criterion from Kilbertus et al. (2017). The problem remains that the causal model needs to be constructed and permissible variables need to be identified manually.

Many more works have been published in this area (e. g., Kilbertus et al., 2019; Wu et al., 2019; Creager et al., 2020b), and it remains an active topic.

2.5 Ground-truth-centric view of bias

The fairness-accuracy trade-off which results from fairness constraints like DP and EOpp has often been seen as a thorny issue. Wick et al. (2019) were one of the first to argue for a change of perspective to solve this. For example, the labels of the ProPublica/COMPAS (Angwin et al., 2016) dataset had been treated as unbiased ground truth in previous work, which the authors call into question. The simple change they do is evaluating fairness-enforcing models on actual *unbiased* test data. In practice, they use simulated datasets such that unbiased labels can be used for evaluation, which is in contrast to real data where we do not have access to ground truth labels, and cannot evaluate on them. The paper considers label bias and "selection bias" ("selecting a subsample of the data in such a way that happens to introduce unexpected correlations, say, between a protected attribute and the target label"), which we previously referred to as sample bias. In their experiments they find, that enforcing fairness can in some situations improve the accuracy on the (unbiased) test set.

Kim et al. (2019) also consider the problem of learning from biased data (and evaluating on *un*biased data). While they do not formulate the problem as a fairness problem, it is equivalent to enforcing fairness in the presence of severe sampling bias. Their main motivating example is the Coloured MNIST dataset; a dataset derived from the MNIST dataset (LeCun et al., 1994). In this dataset, digits are randomly coloured in the test set, but in the training set there is a one-to-one correspondence between digit class and colour. (See figure 2.2 for an example of this dataset.) A naïve classifier will learn to predict colour instead of digit class. The proposed method is very similar to Ganin et al. (2016) and Edwards and Storkey (2016), if we think of the colours as forming different domains (i. e., the *red* domain contains all red digits, etc.). The goal is then to learn a domain-independent representation. The main difference to previous work is an additional entropy loss term in the adversarial loss.

Arjovsky et al. (2019) tackle a similar problem, but take a very different approach, which they term "Invariant Risk Minimisation", contrasted to the usual approach of Empirical Risk Minimisation (ERM). As with Kim et al. (2019), the goal is to ignore spurious correlations that only appear in the very imperfect training set, but not in the test set; and they also perform experiments on Coloured MNIST. They formalise the problem as one of different environments $e \in E$, where each environment is a different, biased view of the same underlying data distribution. The idea is to train a predictor that is simultaneously optimal for all the environments, with the expectation that this generalises to the test set. This idea is formulated as a (intractable) bi-level, constrained optimisation problem, which they approximate with a tractable regularised optimisation.

As the authors point out, the goal of machine learning should be to identify natural, deep, robust structures in reality, instead of relying on superficial correlations. (This dichotomy is sometimes framed as *correlation* vs *causality*, where "causality" takes on a very broad meaning that encompasses any kind of fundamental structure in reality; see also the discussion of *causality* in chapter 1.) Under this view, methods for invariance learning can be framed as trying to learn a more fundamental structure than the dataset at first seems to show. By aiming to be invariant to specific environments/subgroups/domains (denoted by the variable s), they can be seen as making the claim that s does not represent a natural structure, but is merely an artefact of the data generation process. Furthermore, the ground-truth-centric view of dataset bias can be understood as aiming to identify the fundamental structures and aiming to be invariant to everything else. This also fits with Kim et al. (2019)'s work

on Coloured MNIST: colour is not regarded as the fundamental structure in the data.

Creager et al. (2020a) build directly on Arjovsky et al. (2019). They aim to address the limitation that the environments – that one wants to be invariant to – have to be pre-defined. Their contribution is to infer these environments instead, based on identifying which environment splits would most negatively affect an ERM classifier. Their experiments show that this can even improve upon human-designated environments; experiments are performed on a synthetic dataset and Coloured MNIST. In contrast to Kim et al. (2019) and Arjovsky et al. (2019), the authors explicitly point out the connection to traditional fairness methods.

A conceptually very influential work was Friedler et al. (2016). While this work does not explicitly present a ground-truth-centric approach to bias, their concepts of the "construct space" and the "observed space" are close to the idea of the (biased) training distribution and the true underlying distribution.

Jiang and Nachum (2020) also formulate their problem this way: in their setting, the underlying unbiased labels become corrupted by a biased labeller, which results in a biased training set. Their goal is to train a classifier that makes correct predictions consistent with the true labels. However, they treat the true labels as completely unknown and evaluate their models w.r.t. the common fairness definitions on a test set that is just as biased as the training set.

Kallus and Zhou (2018) is another more conceptual work. They present an intuitive model for how sampling bias can enter a training set and how it can be very hard to correct for such a bias. The paper refers to the process as *systematic censoring*, but this is not meant to necessarily imply that this censoring was a conscious decision by some authority; it can also be an unintended side effect of an enacted policy. *Systematic censoring* can arise any time a screening process prevents observing the outcome for the screened-out samples. For example, if, historically, a certain demographic group was screened out from receiving loans, then it was not possible to observe default rates for this group; and so any historical data on defaulting is useless for that demographic group. Even if a bank wants to change their policy, it is hard to correct for the bias in the training set because there is no good data to learn from. This is a prime example of a ground-truth-centric bias problem, as defined in the beginning of the chapter.

Finally, as mentioned in the beginning, Blum and Stangl (2020) provide a formalisation of a setting with label bias and sampling bias and an unbiased ground-truth test set. They explicitly state not being affected by a fairness-

accuracy trade-off, because the true distribution in the considered problem satisfies demographic parity (DP). Their formalised dataset generation process has multiple steps: First, a small amount of random noise is applied to the (true) labels; this models general inaccuracies in the labelling process and does not introduce bias yet. Second, sampling bias is added, by dropping samples based on what the sample's s and y values are. Concretely, only samples with $s = 0$ are dropped. Finally, labelling bias is added, by flipping labels for one of the demographic groups (e. g., $s = 0$), in one specific direction (from $y = 1$ to $y = 0$). The authors show that in this specific situation, an unbiased classifier (as determined by evaluation on the unbiased test set) can be recovered by enforcing its predictions to be compatible with equality of opportunity (EOpp) on the training set. Notably, the fairness constraint enforced during training (EOpp) is not the fairness constraint satisfied by the predictions on the unbiased test set (DP). One reason, why enforcing EOpp works so well here is that the label bias only affects the *underrepresented* demographic group, and that the label bias is the direction ($y = 1$ to $y = 0$) that EOpp is most sensitive to. Thus, even in the ground-truth-centric view of bias, fairness constraints can be very useful.

Maity et al. (2020) provide a similar analysis to Blum and Stangl (2020), but consider more general dataset biases. Just like Blum and Stangl (2020), they explicitly reject a fairness-accuracy trade-off, because they evaluate on a balanced test set. (In the paper this is formulated in terms of the optimal Bayes classifier satisfying their notions of fairness on the test set.) They show that the sampling bias (in the training set) that they consider, can be overcome by enforcing an appropriately chosen risk-based notion of algorithmic fairness; either a notion they term *risk parity* (inspired by DP) or *conditional risk parity* (inspired by EOdds).

A perceived shortcoming of most methods in this area is that they rely on annotations for the subgroups. Two papers that tried to circumvent this problem are Hashimoto et al. (2018) and Nam et al. (2020), both taking very different routes. The first one tries to be robust with respect to all possible subgroups down to a certain size. Essentially, the method tackles the worst-case scenario where the most egregiously misclassified samples form a subgroup. The second approach is based on the idea that subgroups are easier to predict than the actual prediction targets. After all, if this were not the case, then there would not really be a problem. Thus, a classifier is learned in such a way that it predominantly tries to make "easy" predictions, that is, samples where the model is very confident receive a higher weight. The assumption is that this model learns to predict the subgroup (or spurious attribute), for which no labels are available. A second model is then trained,

such that those samples are downweighted for which the first model was very confident; the hope being that by learning the "hard" samples, the model learns the correct relationship. Both Hashimoto et al. (2018) and Nam et al. (2020) frame their methods as improving accuracy on an unbiased test set and do not make references to fairness metrics, but a strong connection to the fairness literature nevertheless exists.

3 | SUMMARY OF CONTRIBUTIONS

The following is a summary of the main contributions in this thesis. All presented approaches deal with dataset bias that is closely linked to a special attribute s. In terms of Kamishima et al. (2012)'s taxonomy (see section section 2.2.3), this can be described as tackling *negative legacy* that is mediated by *prejudice*. Furthermore, the approaches all rely on some form of side information which allows us to overcome dataset bias. This side information is always significantly easier to obtain than unbiased data.

3.1 Mitigating label bias with target labels

The first work is Kehrenberg et al. (2020a) (chapter 4) which is predominantly concerned with label bias. More precisely, labels y are flipped with a probability and a direction that depends on a sensitive attribute s. The main idea is that we make use of pseudo labels (or *target labels*) to implicitly learn from a *balanced* dataset, in which $y \perp s$ holds, and which thus satisfies demographic parity (DP). This falls under the area of *fair classifiers* discussed in section 2.2.3

We can interpret the contributions of this publication in two ways. The first corresponds to the definition-centric view of dataset bias, and the second to the ground-truth-centric view.

1. We can say that the classifier should satisfy demographic parity in its predictions, and learning from a balanced training set is just one particular way to achieve this. In this view, the pseudo labels have no deeper meaning and are just a computational trick.

2. We can see the training set as a corrupted version of a true dataset, which is balanced ($y \perp s$), and so, by learning from these pseudo labels, we are simply approximating the true dataset. However, we do not actually have access to the true dataset; we only know that it is balanced. In order to evaluate the trained model, we compute fairness metrics with respect to DP.

Within the paper, we sometimes jump between these two views.

While the main focus is on label bias, the experiments are performed on real-world fairness datasets, which also display a significant amount of

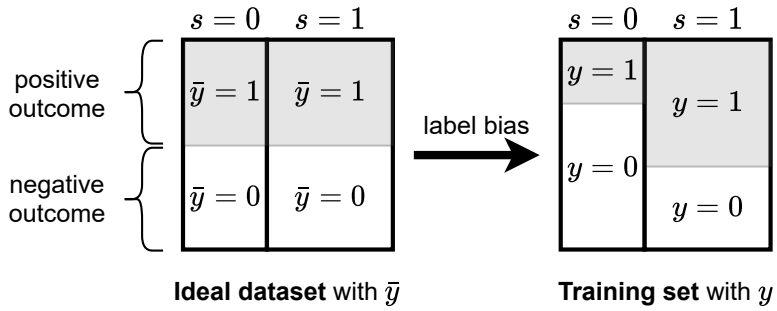


Figure 3.1: A diagram of a simple case of label bias, where both the class label y and the sensitive attribute s are binary. On the left, we have the ideal (possibly fictional) dataset with the target labels $\bar{y}$, where the proportion of positive outcomes is the same for both demographic groups; and on the right, the training labels, for which the proportions are *not* the same.

sample bias (disadvantaged groups are underrepresented). Furthermore, in addition to the result for DP, we also show that the proposed scheme improves equality of opportunity (EOpp).

In order to construct the target labels, we use side information about summary statistics for a balanced training set. This allows us to target a specific balanced set, instead of just any balanced set. In other words, rather than just enforcing DP, the method gives control over the target rates $P(\hat{y}=1|s)$, which are the only fairness-hyperparameters of the model. The target labels $\bar{y}$ represent an uncertain estimate of labels corresponding to a balanced dataset (one where $\bar{y} \perp s$; see also figure 3.1). The idea is then the learn a predictor for these target labels instead of the given (biased) training labels y. Via the sum rule of probabilities, it is possible to express the model likelihood in terms of the target labels, such that maximising the likelihood corresponds to improving the prediction of the target labels.

The requirements for the model are that it outputs probabilities and that they are well-calibrated – which means that for those samples that the model predicts a 10% chance of having a positive label ($y=1$) about 10% in fact have a positive label, and analogously for all other predicted probabilities. The probabilities are needed for calculating the expected target label, and the calibration ensures that this expectation is sensible. Thus, we picked a Gaussian process (GP) model as one model for the experiments, as they have a reputation for being well-calibrated. However, they come with the downside that they are (at least in their standard form) not well-suited to very high dimensional data like images. As such, we also construct a model based on Logistic Regression (LR).

The method is validated with experiments on the UCI Adult Income dataset and the ProPublica/COMPAS dataset, which have been mentioned several

times in chapter 2, and which are the most common tabular fairness datasets. While both these datasets comprise only tabular data, there is nothing in principle that stops this method from being used for other kinds of data. The choice to use these datasets and not others was predominantly made for easier comparison to baselines, and shorter experiment runtime.

3.2 Overcoming severe sampling bias with a representative set

In the setting from the above paper (Kehrenberg et al., 2020a), labels were untrustworthy because they had been flipped; a phenomenon we referred to as *label bias*. However, flipping labels is not the only way that labels can become untrustworthy. Another way is *sampling bias*, which is the subject of Kehrenberg et al. (2020b) and Kehrenberg et al. (2021) (chapters 5 and 6).

As an example, consider the scenario where someone wants to create a classifier to distinguish between sheep and cows, that is supposed to work anywhere on earth. However, they take a shortcut while creating the dataset and take all their sheep images from hot and dry countries and all their cow images from mild and rainy countries. In this case, the dataset is lacking cow images in dry landscapes, and is lacking sheep from green landscapes; the dataset exhibits a strong sampling bias. The result is that even though the labels correctly correspond to cows and sheep, they do not point reliably to the right target anymore. As background colour is easier to recognise with a Convolutional Neural Network (CNN) than animal species, the labels have effectively been turned into landscape labels. In other words, landscape has become a *spurious attribute*. In the following, we denote the spurious attribute with s, as it takes on a role that is very similar to that of the sensitive attribute that was also denoted by s. However, there is a difference in emphasis between a *sensitive* and a *spurious* attribute: the former indicates that the attribute should not be used for legal or ethical reasons, whereas the latter can be any attribute that is associated with the class label y in an undesired way that leads to lower quality generalisation.

The method, proposed in the previous paper (chapter 4), is not able to deal with such a dataset bias as we can easily see: Say, *smiling* corresponds to $y = 1$ and *not smiling* to $y = 0$; furthermore, let red hair correspond to $s = 1$, black hair to $s = 0$, and all other hair colours to $s = 2$. Then, the problem with the described dataset is, that it mostly consists of samples with $y = 0 \wedge s = 0$ and those with $y = 1 \wedge s = 1$. If we call $P(y = 1|s = s')$ the acceptance rate, then the problem can be described as one of very different acceptance rates in the hair colour groups given by s. This is the problem tackled in the previous

paper, and yet, if we were to equalise the acceptance rates with the method there, the result would be very incorrect. The issue is that we would treat the labels as incorrect, when in truth, they are correct; the problem with the data being sampling bias.

To deal with sampling bias, a different approach is needed. Indeed, the problem, as posed, is not solvable in the general case. To make headway with this problem, we introduce the concept of a *representative set*. This set is not subject to the sampling bias, but is unlabelled (with respect to y) and so does not by itself suffice for training. However, this set does have labels for the spurious attribute s. This allows us to learn an *invariant representation*, i. e., a representation of the input features x which is invariant to the spurious attribute. This kind of representation is equivalent to a fair representation – as described in section 2.2.5 – which is invariant to a sensitive attribute. With the invariant representation of the training set, a classifier can then be trained to accurately predict the class label y. The invariant representation cannot be learned from the training set because there, due to the sampling bias, s and y are not sufficiently distinguishable.

A parallel to the previous paper is that the method makes use of side information (in this case the representative set) in order to overcome the bias in the training set.

The method implementing this general strategy, and presented in Kehrenberg et al. (2020b) (chapter 5), is based on the idea of *null-sampling*, which refers to zeroing out part of an encoding, and then reconstructing the modified encoding as if it were a normal encoding. In order to apply null-sampling, an encoding of the input x is learned that is split into two parts: z_u, which has no information about the spurious attribute s, and z_b, which has all the remaining information needed to reconstruct x that is not contained in z_u. z_u is ensured to be not predictive of s via adversarial training. During null-sampling, z_b is zeroed out, and after decoding it, we obtain an invariant representation *in the data domain*. The fact that it is in the data domain makes it interpretable (or inspectable) as defined in chapter 1.

The described method works particularly well with Invertible Neural Networks (INNs), as they ensure that no information is lost that is unrelated to s. However, the price to pay for using INNs is higher memory requirement and slower training. Thus, we also present a variant of the method using a VAE, which does not have the guarantee about preserving information, but also does not suffer from the increased training cost as much. VAEs are similar to INNs in that their encoding conforms to a specific probability distribution, from which we can sample our null-samples. The choice between the two presented variants is determined by whether the user is willing to accept

higher training costs for a lower probability of losing information needed for any prediction tasks. However, the key element is simply any kind of encoder – producing a split-encoding – whose output can be subjected to adversarial training, so encoders other than VAEs or INNs will potentially work as well.

We perform experiments on the Coloured MNIST dataset (as described in section 2.5), which has a one-to-one mapping between the class label (i. e., digit) and the spurious attribute (colour) in the training set. As colour is "easier" to learn, a neural network will learn to predict s instead of y. For additional experiments on the CelebA dataset (another image dataset) and the UCI Adult Income dataset (a tabular dataset), we deliberately apply sampling bias to the training set and then apply our method. For the tabular dataset, an autoencoder is trained to produce a continuous representation, which is then fed into the INN or VAE model. Thus, we demonstrate that the method is *general* in the sense that it is applicable to both image-based and tabular datasets and furthermore should be applicable to other modalities as well. For the main experiments, we focus on image datasets, because they are easiest to visualise in a document.

3.3 Overcoming sampling bias with an unlabelled deployment set

A shortcoming of the approach from the previous publication (Kehrenberg et al., 2020b) is its reliance on a representative set which has labels for the spurious attribute s. As discussed, it is necessary to make use of *some* kind of side information, but perhaps we can relax some requirements. In particular, while it is already easier to collect data without y labels (but with s labels), it is even easier to collect data without *any* label. Thus, requiring only an unlabelled context set would improve the applicability of the method. Kehrenberg et al. (2021) (chapter 6) presents an approach based on that idea.

The setting is very similar to the previous publication (Kehrenberg et al., 2020b): the training set suffers from severe sampling bias, but we have access to a (mostly) unbiased *deployment set* (similar to but not quite identical to the previously discussed *representative set*). The idea is that the deployment set corresponds to the setting in which the model is meant to be deployed. The change from the previous setting is that this additional set may be completely unlabelled, but in exchange, we have some stronger requirements for the training set: The holes left by the sampling bias may not be so numerous as to make s and y completely indistinguishable. For example, in the previous work, the example of Coloured MNIST had a training set where there was a strict one-to-one mapping of colour and digit; but this kind of blending of s

and y into one is not the focus of this paper. Instead, the focus is on a setting where the training set lacks certain combinations of s and y, which results in poor predictions for these combinations on the test set (or deployment setting) where those combinations *do* occur. We refer to these missing combinations as *subgroup bias* or *missing subgroups*, depending on whether a given s value appears in the training set at all. The label s plays here a similar role to the spurious attribute in the previous publication, but as there is a change in emphasis, we refer to s as *subgroup label*[1] instead. The difference between a spurious attribute and a subgroup label is that the former is mostly characterised via its confusion with the prediction target, whereas the latter refers to natural groups in the data which have differing levels of annotation quality which affects the classification performance on these subgroups. However, in both cases, the goal is to make the model output invariant to the s label, i. e.the classification performance should be independent of the subgroup.

As in the previous paper, the first step is to train a neural network to produce an invariant representation. The second step is then to train a classifier on said representation. The invariant representation is trained by performing *distribution matching* between the training set and the deployment set.

The distribution matching is realised with adversarial networks which compare batches of samples, in order to try to distinguish data drawn from the training set and the deployment set. This process requires balanced batches as an inductive bias, because the network will only learn the intended difference between training and deployment set, if the drawn batches exhibit this difference. For example, if batches drawn from Coloured MNIST differ not in colour but in digit class, then distribution matching will learn to change digit shapes. Balancing batches from the training set is easily possible with the available labels, but those are not available for the deployment set, so we use clustering techniques to identify the different groups in the deployment set, and then draw samples for the batches at an equal rate from all clusters. It is important to note here that imperfections in the clustering are not a problem as long as the batches show on average the intended difference between training and deployment set.

The absolutely essential elements for this method are the encoder (also referred to as 'de-biaser') that encodes both, samples from the training set and samples from the deployment set, to a splittable representation; the adversary that tries to identify, from one part of the split representation, which dataset the given samples originated from; and some kind of reconstruction loss to ensure the splittable representation represents the input data well. The

1 This terminology is inspired by the *subclass* concept in Sohoni et al. (2020).

following elements are in theory optional, but are needed for the method to work with real-world data: clustering and sampling to ensure that the training batches are balanced in specific ways; subdivision of the batches into bags; and the aggregation of the adversarial loss over the bags. Roughly speaking, these elements strengthen the supervision signal for the invariance learning.

Experiments are performed on the same datasets as in the previous papers: Coloured MNIST, UCI Adult Income and CelebA. Again, these datasets were chosen, because image data is easy to visualise, and because demonstrating the method on tabular data provides evidence for the generality.

3.4 List of publications and author contributions

This thesis is based on 3 publications (one of which is a work in progress), corresponding to chapters 4–6. The following is a detailed listing of all the individual author contributions.

3.4.1 *Publication 1*

Kehrenberg, Thomas, Zexun Chen and Novi Quadrianto (2020). 'Tuning Fairness by Balancing Target Labels'. In: *Frontiers in Artificial Intelligence* 3, p. 33. DOI: 10.3389/frai.2020.00033.

A shorter version was published as a workshop paper:

Kehrenberg, Thomas, Zexun Chen and Novi Quadrianto (2018). 'Interpretable Fairness via Target Labels in Gaussian Process Models'. In: *Workshop on Ethical, Social and Governance Issues in AI at NeurIPS*. Montreal, Canada.

Contributions:

- I conceived the idea of using Target Labels to target a balanced set. I developed the proof from a starting point that my supervisor pointed me to. I wrote all of the code dealing with the modified loss function and nearly all of the remaining code as well. I ran most of the experiments and wrote all of the methods section and most of the remaining text as well.
- Z. Chen was a discussion partner and helped run the experiments, and wrote some parts of the code.
- N. Quadrianto suggested the initial direction of the work, was a discussion partner, and helped write the introduction and related work.

3.4.2 *Publication 2*

Kehrenberg, Thomas, Myles Bartlett, Oliver Thomas and Novi Quadrianto (2020). 'Null-sampling for Interpretable and Fair Representations'. In: *European Conference on Computer Vision (ECCV)*. Glasgow, UK. DOI: 10.1007/978-3-030-58604-1.

CONTRIBUTIONS:

- I conceived the idea of using an Invertible Neural Network (INN) and a representative set to learn an invariant representation. I wrote a large part of the code, and ran about half the experiments. I wrote a significant part of the text.
- M. Bartlett developed a large part of the tricks needed for training the INN successfully. He wrote a significant part of the code, and of the text.
- O. Thomas helped with writing the code and with running the experiments.
- N. Quadrianto gave feedback on the progress and suggested directions to explore.

3.4.3 *Publication 3 (work in progress)*

Kehrenberg, Thomas, Viktoriia Sharmanska, Myles Bartlett and Novi Quadrianto (2021). 'Learning with Perfect Bags: Addressing Hidden Stratification with Zero Labeled Data'.

CONTRIBUTIONS:

- I developed the idea of distribution matching as an extension of the previous paper. In addition, I tried to achieve similar goals by applying clustering to an unlabelled auxiliary set, with supervision from the labelled training set. I wrote all of the initial implementation of the method, and a large part of the later refined implementation. I ran the majority of the experiments.
- V. Sharmanska developed the original link to a fairness problem and wrote parts of the introduction and related work and contributed to other sections.

- M. Bartlett introduced the idea of batches-of-bags. He also improved the NN architectures of the encoder and the discriminator, ran experiments, wrote the sections on architecture and contributed to other parts of the paper.
- N. Quadrianto suggested how to combine the two research directions that I had into a coherent whole. He also gave general feedback and suggested directions to explore.

Part II

PUBLICATIONS

This part comprises two peer-reviewed publications and one work in progress. They are reproduced here with minimal changes.

4 PAPER 1: TUNING FAIRNESS BY BALANCING TARGET LABELS

AUTHORS: Thomas Kehrenberg[1], Zexun Chen[1,2], and Novi Quadrianto[1]
AFFILIATIONS:
[1] Predictive Analytics Lab (PAL), University of Sussex, Brighton, UK
[2] BioComplex Laboratory, University of Exeter, Exeter, UK
JOURNAL: *Frontiers in Artificial Intelligence*, Volume 3
DOI: 10.3389/frai.2020.00033
NOTE: The appendix has been included as section 4.8.

4.1 ABSTRACT

The issue of fairness in machine learning models has recently attracted a lot of attention as ensuring it will ensure continued confidence of the general public in the deployment of machine learning systems. We focus on mitigating the harm incurred by a biased machine learning system that offers better outputs (e.g. loans, job interviews) for certain groups than for others. We show that bias in the output can naturally be controlled in probabilistic models by introducing a latent target output. This formulation has several advantages: first, it is a unified framework for several notions of group fairness such as Demographic Parity and Equality of Opportunity; second, it is expressed as a marginalisation instead of a constrained problem; and third, it allows the encoding of our knowledge of what unbiased outputs should be. Practically, the second allows us to avoid unstable constrained optimisation procedures and to reuse off-the-shelf toolboxes. The latter translates to the ability to control the level of fairness by directly varying fairness target rates. In contrast, existing approaches rely on intermediate, arguably unintuitive, control parameters such as covariance thresholds.

4.2 INTRODUCTION

Algorithmic assessment methods are used for predicting human outcomes in areas such as financial services, recruitment, crime and justice, and local government. This contributes, in theory, to a world with decreasing human biases. To achieve this, however, we need fair machine learning models that

take biased datasets, but output non-discriminatory decisions to people with differing protected attributes such as gender and marital status. Datasets can be biased because of, for example, sampling bias, subjective bias of individuals, and institutionalised biases (Olteanu et al., 2019; Tolan, 2019). Uncontrolled bias in the data can translate into bias in machine learning models.

There is no single accepted definition of algorithmic fairness for automated decision-making but several have been proposed. One definition is referred to as *statistical* or *demographic parity*. Given a binary protected attribute (e.g. married/unmarried) and a binary decision (e.g. yes/no to getting a loan), demographic parity requires equal positive rates (PR) across the two sensitive groups (married and *un*married individuals should be equally likely to receive a loan). Another fairness criterion, *equalised odds* (Hardt et al., 2016), takes into account the binary decision, and instead of equal PR requires equal true positive rates (TPR) and false positive rates (FPR). This criterion is intended to be more compatible with the goal of building accurate predictors or achieving high utility (Hardt et al., 2016). We discuss the suitability of the different fairness criteria in the discussion section at the end of the paper.

There are many existing models for enforcing demographic parity and equalised odds (Calders et al., 2009; Kamishima et al., 2012; Zafar et al., 2017a,b; Agarwal et al., 2018; Creager et al., 2019). However, these existing approaches to balancing accuracy and fairness rely on intermediate, unintuitive control parameters such as allowable constraint violation ϵ (e.g. 0.01) in Agarwal et al. (2018), or a covariance threshold c (e.g. 0 that is controlled by other parameters τ and μ – 0.005 and 1.2 – to trade off this threshold and accuracy) in Zafar et al. (2017a). This is related to the fact that many of these approaches embed fairness criteria as *constraints* in the optimisation procedure (Quadrianto and Sharmanska, 2017; Zafar et al., 2017a,b; Donini et al., 2018).

In contrast, we provide a probabilistic classification framework with bias controlling mechanisms that can be tuned based on positive rates (PR), an intuitive parameter. Thus, giving humans the control to set the rate of positive predictions (e.g. a PR of 0.6). Our framework is based on the concept of a *balanced dataset* and introduces latent target labels, which, instead of the provided labels, are now the training label of our classifier. We prove bounds on how far the target labels diverge from the dataset labels. We instantiate our approach with a parametric logistic regression classifier and a Bayesian non-parametric Gaussian process classifier (GPC). As our formulation is not expressed as a constrained problem, we can draw upon advancements in automated variational inference (Bonilla et al., 2016; Krauth et al., 2017;

Gardner et al., 2018) for learning the fair model, and for handling large amounts of data.

The method presented in this paper is closely related to a number of previous works, e.g. Calders and Verwer (2010) and Kamiran and Calders (2012). Proper comparison with them requires knowledge of our approach. We will thus explain our approach in the subsequent sections, and defer detailed comparisons to section 4.5 (Related Work).

4.3 Target labels for tuning group fairness

We will start by describing several notions of group fairness. For each individual, we have a vector of non-sensitive attributes $x \in \mathcal{X}$, a class label $y \in \mathcal{Y}$, and a sensitive attribute $s \in \mathcal{S}$ (e.g. racial origin or gender). We focus on the case where s and y are binary. We assume that a positive label $y = 1$ corresponds to a positive outcome for an individual – for example, being accepted for a loan. *Group fairness* balances a certain condition between groups of individuals with different sensitive attributes, s versus s'. The term $\hat{y}$ below is the prediction of a machine learning model that, in most works, uses only non-sensitive attributes x. Several group fairness criteria have been proposed (e.g. Hardt et al., 2016; Chouldechova, 2017; Zafar et al., 2017a):

equality of positive rate (Demographic Parity):

$$P(\hat{y} = 1|s) = P(\hat{y} = 1|s') \tag{4.1}$$

equality of accuracy:

$$P(\hat{y} = y|s) = P(\hat{y} = y|s') \tag{4.2}$$

equality of true positive rate (Equality of Opportunity):

$$P(\hat{y} = 1|s, y = 1) = P(\hat{y} = 1|s', y = 1)\,. \tag{4.3}$$

Equalised odds criterion corresponds to Equality of Opportunity (4.3) plus equality of false positive rate.

The Bayes-optimal classifier only satisfies these criteria if the training data itself satisfies them. That is, in order for the Bayes-optimal classifier to satisfy *demographic parity*, the following must hold: $P(y = 1|s) = P(y = 1|s')$, where y is the training label. We call a dataset for which $P(y, s) = P(y)P(s)$ holds, a *balanced* dataset. Given a balanced dataset, a Bayes-optimal classifier learns to satisfy demographic parity and an approximately Bayes-optimal classifier should learn to satisfy it at least approximately. Here, we motivated the importance of balanced datasets via the demographic parity criterion, but it is also important for *equality of opportunity* which we discuss in Section 4.3.1.

In general, however, our given dataset is likely to be imbalanced. There are two common solutions to this problem: either pre-process or massage the dataset to make it balanced, or constrain the classifier to give fair predictions despite it having been trained on an unbalanced dataset. Our approach takes parts from both solutions.

An imbalanced dataset can be turned into a balanced dataset by either changing the class labels y or the sensitive attributes s. In the use cases that we are interested in, s is considered an integral part of the input, representing trustworthy information and thus should not be changed. y, conversely, is often not completely trustworthy; it is not an integral part of the sample but merely an observed outcome. In a hiring dataset, for instance, y might represent the hiring decision, which can be biased, and not the relevant question of whether someone makes a good employee.

Thus, we introduce new *target labels* $\bar{y}$ such that the dataset is balanced: $P(\bar{y}, s) = P(\bar{y})P(s)$. The idea is that these target labels still contain as much information as possible about the task, while also forming a balanced dataset. This introduces the concept of the accuracy-fairness trade-off: in order to be completely accurate with respect to the original (not completely trustworthy) class labels y, we would require $\bar{y} = y$, but then, the fairness constraints would not be satisfied.

Let $\eta_s(x) = P(y = 1|x, s)$ denote the distribution of y in the data. The target distribution $\bar{\eta}_s(x) = P(\bar{y} = 1|x, s)$ is then given by

$$\begin{aligned}\bar{\eta}_s(x) = &(P(\bar{y} = 1|y = 1, s) + P(\bar{y} = 0|y = 0, s) - 1) \cdot \eta_s(x) \\ &+ 1 - P(\bar{y} = 0|y = 0, s)\end{aligned} \tag{4.4}$$

due to the marginalisation rules of probabilities. The conditional probability $P(\bar{y}|y, s)$ indicates with which probability we want to keep the class label. This probability could in principle depend on x which would enable the realisation of individual fairness. The dependence on x has to be prior knowledge as it cannot be learned from the data. This prior knowledge can encode the semantics that "similar individuals should be treated similarly" (Dwork et al., 2012), or that "less qualified individuals should not be preferentially favoured over more qualified individuals" (Joseph et al., 2016). Existing proposals for guaranteeing individual fairness require strong assumptions, such as the availability of an agreed-upon similarity metric, or knowledge of the underlying data generating process. In contrast, in group fairness, we partition individuals into protected groups based on some sensitive attribute s and ask that some statistics of a classifier be approximately equalised across those groups (see (4.1)–(4.3)). In this case, $P(\bar{y}|y, s)$ does not depend on x.

Returning to equation 4.4, we can simplify it with

$$m_s := P(\bar{y} = 1|y = 1, s) + P(\bar{y} = 0|y = 0, s) - 1 \tag{4.5}$$

$$b_s := 1 - P(\bar{y} = 0|y = 0, s)\,, \tag{4.6}$$

arriving at $\bar{\eta}_s(x) = m_s \cdot \eta_s(x) + b_s$. m_s and b_s are chosen such that $P(\bar{y}, s) = P(\bar{y})P(s)$. This can be interpreted as shifting the decision boundary depending on s so that the new distribution is balanced.

As there is some freedom in choosing m_s and b_s, it is important to consider what the effect of different values is. The following theorem provides this (the proof can be found in the Supplementary Material):

THEOREM 4.1. *The probability that y and $\bar{y}$ disagree ($y \neq \bar{y}$) for any input x in the dataset is given by:*

$$P(y \neq \bar{y}|s) = P\Big(\Big|\eta(x, s) - \tfrac{1}{2}\Big| < t_s\Big) \tag{4.7}$$

where

$$t_s = \left|\frac{m_s + 2b_s - 1}{2m_s}\right|\,. \tag{4.8}$$

Thus, if the threshold t_s is small, then only if there are inputs very close to the decision boundary ($\eta_s(x)$ close to $\frac{1}{2}$) would we have $\bar{y} \neq y$. t_s determines the accuracy penalty that we have to accept in order to gain fairness. The value of t_s can be taken into account when choosing m_s and b_s (see section 4.4). If η_s satisfies the Tsybakov condition (Tsybakov et al., 2004), then we can give an upper bound for the probability.

DEFINITION 4.1. A distribution η satisfies the Tsybakov condition if there exist $C > 0$, $\lambda > 0$ and $t_0 \in (0, \frac{1}{2}]$ such that for all $t \leq t_0$,

$$P\Big(\Big|\eta(x) - \tfrac{1}{2}\Big| < t\Big) \leq Ct^{\lambda}\,. \tag{4.9}$$

This condition bounds the region close to the decision boundary. It is a property of the dataset.

COROLLARY 4.1.1. *If $\eta(x, s) = P(y = 1|x, s)$ satisfies the Tsybakov condition in x, with constants C and λ, then the probability that y and $\bar{y}$ disagree ($y \neq \bar{y}$) for any input x in the dataset is bounded by:*

$$P(y \neq \bar{y}|s) < C\left|\frac{m_s + 2b_s - 1}{2m_s}\right|^{\lambda}\,. \tag{4.10}$$

Section 4.4 discusses how to choose the parameters for $\bar{\eta}$ in order to make it balanced.

4.3.1 *Equality of Opportunity*

In contrast to demographic parity, equality of opportunity (just as equality of accuracy) is satisfied by a perfect classifier. Imperfect classifiers, however, do not by default satisfy it: the true positive rate (TPR) is different for different subgroups. The reason for this is that while the classifier is optimised to have a high TPR overall, it is not optimised to have the same TPR in the subgroups.

The overall TPR is a weighted sum of the TPRs in the subgroups:

$$TPR = P(s = 0|y = 1) \cdot TPR_{s=0} + P(s = 1|y = 1) \cdot TPR_{s=1} \,. \tag{4.11}$$

In datasets where the positive label $y = 1$ is heavily skewed toward one of the groups (say, group $s = 1$; meaning that $P(s = 1|y = 1)$ is high and $P(s = 0|y = 1)$ is low), overall TPR might be maximised by setting the decision boundary such that nearly all samples in $s = 0$ are classified as $y = 0$, while for $s = 1$ a high TPR is achieved. The low TPR for $s = 0$ is in this case weighted down and only weakly impacts the overall TPR. For $s = 0$, the resulting classifier uses s as a shorthand for y, mostly ignoring the other features. This problem usually persists even when s is removed from the input features because s is implicit in the other features.

A *balanced* dataset helps with this issue because in such datasets, s is not a useful proxy for the balanced label $\bar{y}$ (because we have $P(\bar{y}, s) = P(\bar{y})P(s)$) and s cannot be used as a shorthand. Assuming the dataset is balanced in s ($P(s = 0) = P(s = 1)$), for such datasets $P(s = 0|y = 1) = P(s = 1|y = 1)$ holds and the two terms in equation 4.11 have equal weight.

Here as well there is an accuracy-fairness trade-off: assuming the unconstrained model is as accurate as its model complexity allows, adding additional constraints like equality of opportunity can only make the accuracy worse.

4.3.2 *Concrete algorithm*

For training, we are only given the unbalanced distribution $\eta_s(x)$ and not the target distribution $\bar{\eta}_s(x)$. However, $\bar{\eta}_s(x)$ is needed in order to train a fair classifier. One approach is to explicitly change the labels y in the dataset, in order to construct $\bar{\eta}_s(x)$. We discuss this approach and its drawback in the related work section (section 4.5).

We present a novel approach which only implicitly constructs the balanced dataset. This framework can be used with any likelihood-based model, such as Logistic Regression and Gaussian Process models. The relation presented in equation 4.4 allows us to formulate a likelihood that targets $\bar{\eta}_s(x)$ while

only having access to the imbalanced labels y. As we only have access to y, $P(y|x,s,\theta)$ is the likelihood to optimise. It represents the probability that y is the imbalanced label, given the input x, the sensitive attribute s that is available in the training set and the model parameters θ for a model that is targeting $\bar{y}$. Thus, we get

$$\begin{aligned} P(y=1|x,s,\theta) &= \sum_{\bar{y}\in\{0,1\}} P(y=1,\bar{y}|x,s,\theta) \\ &= \sum_{\bar{y}\in\{0,1\}} P(y=1|\bar{y},x,s,\theta)\, P(\bar{y}|x,s,\theta)\,. \end{aligned} \tag{4.12}$$

As we are only considering group fairness, we have $P(y=1|\bar{y},x,s,\theta) = P(y=1|\bar{y},s)$.

Let $f_\theta(x,y')$ be the likelihood function of a given model, where f gives the likelihood of the label y' given the input x and the model parameters θ. As we do not want to make use of s at test time, f does not explicitly depend on s. The likelihood with respect to $\bar{y}$ is then given by f: $P(\bar{y}|x,s,\theta) = f_\theta(x,\bar{y})$; and thus, does not depend on s. The latter is important in order to avoid *direct discrimination* (Barocas and Selbst, 2016). With these simplifications, the expression for the likelihood becomes

$$P(y=1|x,s,\theta) = \sum_{\bar{y}\in\{0,1\}} P(y=1|\bar{y},s)\, P(\bar{y}|x,\theta)\,. \tag{4.13}$$

The conditional probabilities, $P(y|\bar{y},s)$, are closely related to the conditional probabilities in equation 4.4 and play a similar role of "transition probabilities". Section 4.4 explains how to choose these transition probabilities in order to arrive at a balanced dataset. For a binary sensitive attribute s (and binary label y), there are 4 transition probabilities (see Algorithm 1 where $d_{\bar{y}=i}^{s=j} := P(y=1|\bar{y}=i,s=j)$):

$$P(y=1|\bar{y}=0,s=0), \qquad P(y=1|\bar{y}=1,s=0) \tag{4.14}$$

$$P(y=1|\bar{y}=0,s=1), \qquad P(y=1|\bar{y}=1,s=1)\,. \tag{4.15}$$

A perhaps useful interpretation of equation 4.13 is that, even though we do not have access to $\bar{y}$ directly, we can still compute the expectation value over the possible values of $\bar{y}$.

The above derivation applies to binary classification but can easily be extended to the multi-class case.

Algorithm 1 Fair learning with target labels $\bar{y}$

INPUT: Training set $\mathscr{D} = \{(x_i, y_i, s_i)\}_{i=1}^N$, transition probabilities $d_{\bar{y}=0}^{s=0}$, $d_{\bar{y}=1}^{s=0}$, $d_{\bar{y}=0}^{s=1}$, $d_{\bar{y}=1}^{s=1}$
OUTPUT: fair model parameters θ

Initialise θ (randomly)
for all x_i, y_i, s_i **do**
  $P_{\bar{y}=1} \leftarrow \bar{\eta}(x_i, \theta)$ (e.g. logistic($\langle x, \theta \rangle$))
  $P_{\bar{y}=0} \leftarrow 1 - P_{\bar{y}=1}$
  if $s_i = 0$ **then**
    $P_{y=1} \leftarrow d_{\bar{y}=0}^{s=0} \cdot P_{\bar{y}=0} + d_{\bar{y}=1}^{s=0} \cdot P_{\bar{y}=1}$
  else
    $P_{y=1} \leftarrow d_{\bar{y}=0}^{s=1} \cdot P_{\bar{y}=0} + d_{\bar{y}=1}^{s=1} \cdot P_{\bar{y}=1}$
  end if
  $\ell \leftarrow y_i \cdot P_{y=1} + (1 - y_i) \cdot (1 - P_{y=1})$
  update θ to maximise likelihood ℓ
end for

4.4 Transition probabilities for a balanced dataset

This section focuses on how to set values of the transition probabilities in order to arrive at balanced datasets.

4.4.1 *Meaning of the parameters*

Before we consider concrete values, we give some intuition for the transition probabilities. Let $s = 0$ refer to the protected group. For this group, we want to make more positive predictions than the training labels indicate. Variable $\bar{y}$ is supposed to be our target proxy label. Thus, in order to make more positive predictions, some of the $y = 0$ labels should be associated with $\bar{y} = 1$. However, we do not know which. So, if our model predicts $\bar{y} = 1$ (high $P(\bar{y} = 1|x, \theta)$) while the training label is $y = 0$, then we allow for the possibility that this is actually correct. That is, $P(y = 0|\bar{y} = 1, s = 0)$ is not 0. If we choose, for example, $P(y = 0|\bar{y} = 1, s = 0) = 0.3$ then that means that 30% of positive target labels $\bar{y} = 1$ may correspond to negative training labels $y = 0$. This way we can have more $\bar{y} = 1$ than $y = 1$, overall. On the other hand, predicting $\bar{y} = 0$ when $y = 1$ holds, will always be deemed incorrect: $P(y = 1|\bar{y} = 0, s = 0) = 0$; this is because we do not want any additional negative labels.

For the non-protected group $s = 1$, we have the exact opposite situation. If anything, we have too many positive labels. So, if our model predicts $\bar{y} = 0$ (high $P(\bar{y} = 0|x, \theta)$) while the training label is $y = 1$, then we should again allow for the possibility that this is actually correct. That is, $P(y = 1|\bar{y} =$

$0, s = 1)$ should not be 0. On the other hand, $P(y = 0|\bar{y} = 1, s = 1)$ should be 0 because we do not want additional positive labels for $s = 1$. It could also be that the number of positive labels is exactly as it should be, in which case we can just set $y = \bar{y}$ for all data points with $s = 1$.

4.4.2 *Choice of parameters*

A balanced dataset is characterised by an independence of the label $\bar{y}$ and the sensitive attribute s. Given that we have complete control over the *transition probabilities*, we can ensure this independence by requiring $P(\bar{y} = 1|s = 0) = P(\bar{y} = 1|s = 1)$. Our constraint is then that both of these probabilities are equal to the same value, which we will call the target rate PR_t ("PR" as *positive rate*):

$$P(\bar{y} = 1|s = 0) \stackrel{!}{=} PR_t \quad \text{and} \quad P(\bar{y} = 1|s = 1) \stackrel{!}{=} PR_t \,. \tag{4.16}$$

This leads us to the following constraints for $s' \in \{0, 1\}$:

$$PR_t = P(\bar{y} = 1|s = s') = \sum_y P(\bar{y} = 1|y, s = s')\, P(y|s = s'). \tag{4.17}$$

We call $P(y = 1|s = j)$ the base rate PR_b^j which we estimate from the training set:

$$P(y = 1|s = i) = \frac{\text{number of points with } y = 1 \text{ in group } i}{\text{number of points in group } i} \,. \tag{4.18}$$

Expanding the sum, we get

$$PR_t = \; P(\bar{y} = 1|y = 0, s = s') \cdot (1 - PR_b^1) + P(\bar{y} = 1|y = 1, s = s') \cdot PR_b^1 \,. \tag{4.19}$$

This is a system of linear equations consisting of two equations (one for each value of s') and four free variables: $P(\bar{y} = 1|y, s)$ with $y, s \in \{0, 1\}$. The two unconstrained degrees of freedom determine how strongly the accuracy will be affected by the fairness constraint. If we set $P(\bar{y} = 1|y = 1, s)$ to 0.5, then this expresses the fact that a train label y of 1 only implies a target label $\bar{y}$ of 1 in 50% of the cases. In order to minimise the effect on accuracy, we make $P(\bar{y} = 1|y = 1, s)$ as high as possible and $P(\bar{y} = 1|y = 0, s)$, conversely, as low as possible. However, the lowest and highest possible values are not

always 0 and 1 respectively. To see this, we solve for $P(\bar{y} = 1|y = 0, s = j)$ in equation 4.19:

$$P(\bar{y} = 1|y = 0, s = j) = \frac{PR_b^j}{1 - PR_b^j}\left(\frac{PR_t}{PR_b^j} - P(\bar{y} = 1|y = 1, s = j)\right). \tag{4.20}$$

If PR_t/PR_b^j were greater than 1, then setting $P(\bar{y} = 1|y = 0, s = j)$ to 0 would imply a $P(\bar{y} = 1|y = 1, s = j)$ value greater than 1. A visualisation that shows why this happens can be found in the Supplementary Material. We thus arrive at the following definitions:

$$P(\bar{y} = 1|y = 1, s = j) = \begin{cases} 1 & \text{if } PR_t > PR_b^j \\ \frac{PR_t}{PR_b^j} & \text{otherwise.} \end{cases} \tag{4.21}$$

$$P(\bar{y} = 1|y = 0, s = j) = \begin{cases} \frac{PR_t - PR_b^j}{1 - PR_b^j} & \text{if } PR_t > PR_b^j \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

Algorithm 2 shows pseudocode of the procedure, including the computation of the allowed minimal and maximal value.

Once all these probabilities have been found, the transition probabilities needed for equation 4.13 are fully determined by applying Bayes' rule:

$$P(y = 1|\bar{y}, s) = \frac{P(\bar{y}|y = 1, s)P(y = 1|s)}{P(\bar{y}|s)}. \tag{4.23}$$

CHOOSING A TARGET RATE. As shown, there is a remaining degree of freedom when targeting a balanced dataset: the target rate $PR_t := P(\bar{y} = 1)$. This is true for both fairness criteria that we are targeting. The choice of targeting rate affects how much η and $\bar{\eta}$ differ as implied by Theorem 4.1 (PR_t affects m_s and b_s). $\bar{\eta}$ should remain close to η as $\bar{\eta}$ only represents an auxiliary distribution that does not have meaning on its own. The threshold t_s in Theorem 4.1 (equation 4.8) gives an indication of how close the distributions are. With the definitions in equation 4.21 and equation 4.22, we can express t_s in terms of the target rate and the base rate:

$$t_s = \begin{cases} \frac{1}{2}\frac{PR_b^s - PR_t}{PR_t} & \text{if } PR_t > PR_b^j \\ \frac{1}{2}\frac{PR_t - PR_b^s}{1 - PR_t} & \text{otherwise.} \end{cases} \tag{4.24}$$

This shows that t_s is smallest when PR_b^s and PR_t are closest. However, as PR_b^s has different values for different s, we cannot set $PR_b^s = PR_t$ for all s. In order to keep both $t_{s=0}$ and $t_{s=1}$ small, it follows from equation 4.24 that PR_t should at least be between PR_b^0 and PR_b^1. A more precise statement can be made when

we explicitly want to minimise the sum $t_{s=0}+t_{s=1}$: assuming $PR_b^0 < PR_t < PR_b^1$ and $PR_b^1 < \frac{1}{2}$, the optimal choice for PR_t is PR_b^1 (see Supplementary Material for details). We call this choice PR_t^{max}. For $PR_b^0 > \frac{1}{2}$, analogous statements can be made, but this is of less interest as this case does not appear in our experiments.

The previous statements about t_s do not directly translate into observable quantities like accuracy if the Tsybakov condition is not satisfied, and even if it is satisfied, the usefulness depends on the constants C and λ. Conversely, the following theorem makes a *generally* applicable statement about the accuracy that can be achieved. Before we get to the theorem, we introduce some notation. We are given a dataset $\mathscr{D} = \{(x_i, y_i)\}_i$, where the x_i are vectors of features and the y_i the corresponding labels. We refer to the tuples (x, y) as the *samples* of the dataset. The number of samples is $N = |\mathscr{D}|$.

We assume binary labels ($y \in \{0, 1\}$) and thus can form the (disjoint) subsets $\mathscr{Y}^0$ and $\mathscr{Y}^1$ with

$$\mathscr{Y}^j = \{(x, y) \in \mathscr{D} | y = j\} \quad \text{with } j \in \{0, 1\}\,. \tag{4.25}$$

Furthermore, we associate each sample with a classification $\hat{y} \in \{0, 1\}$. The task of making the classification $\hat{y} = 0$ or $\hat{y} = 1$ can be understood as sorting each sample from $\mathscr{D}$ into one of two sets: $\mathscr{C}^0$ and $\mathscr{C}^1$, such that $\mathscr{C}^0 \cup \mathscr{C}^1 = \mathscr{D}$ and $\mathscr{C}^0 \cap \mathscr{C}^1 = \emptyset$.

We refer to the set $\mathscr{A} = (\mathscr{C}^0 \cap \mathscr{Y}^0) \cup (\mathscr{C}^1 \cap \mathscr{Y}^1)$ as the set of correct (or accurate) predictions. The *accuracy* is given by $acc = N^{-1} \cdot |\mathscr{A}|$.

DEFINITION 4.2.

$$r_a := \frac{|\mathscr{Y}^1|}{|\mathscr{D}|} = \frac{|\mathscr{Y}^1|}{N} \tag{4.26}$$

is called the *base acceptance rate* of the dataset $\mathscr{D}$.

DEFINITION 4.3.

$$\hat{r}_a = \frac{|\mathscr{C}^1|}{|\mathscr{D}|} = \frac{|\mathscr{C}^1|}{N} \tag{4.27}$$

is called the *predictive acceptance rate* of the predictions.

THEOREM 4.2. *For a dataset with the base rate r_a and corresponding predictions with a predictive acceptance rate of $\hat{r}_a$, the accuracy is limited by*

$$acc \leq 1 - |\hat{r}_a - r_a|\ . \tag{4.28}$$

COROLLARY 4.2.1. *Given a dataset that consists of two subsets $\mathscr{S}_0$ and $\mathscr{S}_1$ ($\mathscr{D} = \mathscr{S}_0 \cup \mathscr{S}_1$) where p is the ratio of $|\mathscr{S}_0|$ to $|\mathscr{D}|$ and given corresponding*

Algorithm 2 Targeting a balanced dataset

INPUT: target rate PR_t, biased acceptance rate PR_b^i
OUTPUT: transition probabilities $d_{\bar{y}=j}^{s=i}$

if $PR_t > PR_b^i$ **then**
  $P(\bar{y} = 1|y = 1, s = i) \leftarrow 1$
else
  $P(\bar{y} = 1|y = 1, s = i) \leftarrow \frac{PR_t}{PR_b^i}$
end if
if j=0 **then**
  $P(\bar{y} = 0|y = 1, s = i) \leftarrow 1 - P(\bar{y} = 1|y = 1, s = i)$
  $d_{\bar{y}=0}^{s=i} \leftarrow \frac{P(\bar{y}=0|y=1,s=i)\cdot PR_b^i}{1-PR_t}$
else if j=1 **then**
  $d_{\bar{y}=1}^{s=i} \leftarrow \frac{P(\bar{y}=1|y=1,s=i)\cdot PR_b^i}{PR_t}$
end if

acceptance rates r_a^0 *and* r_a^1 *and predictions with target rates* $\hat{r}_a^0$ *and* $\hat{r}_a^1$*, the accuracy is limited by*

$$acc \leq 1 - p \cdot \left|\hat{r}_a^0 - r_a^0\right| - (1 - p) \cdot \left|\hat{r}_a^1 - r_a^1\right| \ . \tag{4.29}$$

The proofs are fairly straightforward and can be found in the Supplementary Material.

Corollary 4.2.1 implies that in the common case where group $s = 0$ is disadvantaged ($r_a^0 < r_a^1$) and also underrepresented ($p < \frac{1}{2}$), the highest accuracy under demographic parity can be achieved at $PR_t = r_a^1$ with

$$acc \leq 1 - p \cdot \left(r_a^1 - r_a^0\right) \ . \tag{4.30}$$

However, this means willingly accepting a lower accuracy in the (smaller) subset $\mathcal{S}_0$ that is compensated by a very good accuracy in the (larger) subset $\mathcal{S}_1$. A decidedly "fairer" approach is to aim for the same accuracy in both subsets. This is achieved by using the average of the base acceptance rates for the target rate. As we balance the test set in our experiments, this kind of sacrificing of one demographic group does not work there. We compare the two choices (PR_t^{max} and PR_t^{avg}) in section 4.6.

4.4.3 *Conditionally balanced dataset*

There is a fairness definition related to demographic parity which allows conditioning on "legitimate" risk factors ℓ when considering how equal the

demographic groups are treated (Corbett-Davies et al., 2017). This cleanly translates into balanced datasets which are balanced conditioned on ℓ:

$$P(\bar{y} = 1|\ell = \ell', s = 0) \overset{!}{=} P(\bar{y} = 1|\ell = \ell', s = 1)\,. \tag{4.31}$$

We can interpret this as splitting the data into partitions based on the value of ℓ, where the goal is to have all these partitions be balanced. This can easily be achieved by our method by setting a $PR_t(\ell)$ for each value of ℓ and computing the transition probabilities for each sample depending on ℓ.

4.5 Related work

There are several ways to enforce fairness in machine learning models: as a pre-processing step (Kamiran and Calders, 2012; Zemel et al., 2013; Louizos et al., 2016; Lum and Johndrow, 2016; Chiappa, 2019; Quadrianto et al., 2019), as a post-processing step (Feldman et al., 2015; Hardt et al., 2016), or as a constraint during the learning phase (Calders et al., 2009; Zafar et al., 2017a,b; Donini et al., 2018; Dimitrakakis et al., 2019). Our method enforces fairness during the learning phase (an in-processing approach) but, unlike other approaches, we do not cast fair-learning as a *constrained* optimisation problem. Constrained optimisation requires a customised procedure. In Goh et al. (2016), Zafar et al. (2017b), and Zafar et al. (2017a), suitable majorisation-minimisation/convex-concave procedures (Sriperumbudur and Lanckriet, 2009) were derived. Furthermore, such constrained optimisation approaches may lead to more unstable training, and often yield classifiers with both worse accuracy and more unfair (Cotter et al., 2018).

The approaches most closely related to ours were given by Kamiran and Calders (2012) who present four pre-processing methods: *Suppression*, *Massaging the dataset*, *Reweighing*, and *Sampling*. In our comparison we focus on methods 2, 3 and 4, because the first one simply removes sensitive attributes and those features that are highly correlated with them. All the methods given by Kamiran and Calders (2012) aim only at enforcing demographic parity.

The massaging approach uses a classifier to first rank all samples according to their probability of having a positive label ($y = 1$) and then flips the labels that are closest to the decision boundary such that the data then satisfies demographic parity. This *pre-processing* approach is similar in spirit to our *in-processing* method but differs in the execution. In our method (section 4.4.2), "ranking" and classification happen in one step and labels are not explicitly flipped but assigned probabilities of being flipped.

The reweighting method reweights samples based on whether they belong to an over-represented or under-represented demographic group. The sampling approach is based on the same idea but works by resampling instead of reweighting. Both reweighting and sampling aim to effectively construct a balanced dataset, without affecting the labels. This is in contrast to our method which treats the class labels as potentially untrustworthy and allows defying them.

One approach in Calders and Verwer (2010) is also worth mentioning. It is based on a *generative* Naïve Bayes model in which a latent variable L is introduced which is reminiscent to our target label $\bar{y}$. We provide a *discriminative* version of this approach. In discriminative models, parameters capture the conditional relationship of an output given an input, while in generative models, the joint distribution of input-output is parameterised. With this conditional relationship formulation ($P(y|\bar{y}, s) = {}^{P(\bar{y}|y,s)P(y|s)}/_{P(\bar{y}|s)}$), we can have detailed control in setting the target rate. Calders and Verwer (2010) focuses only on the demographic parity fairness metric.

4.6 Experiments

We compare the performance of our target-label model with other existing models based on two real-world datasets. These datasets have been previously considered in the fairness-aware machine learning literature.

4.6.1 *Implementation*

The proposed method is compatible with any likelihood-based algorithm. We consider both a nonparametric and a parametric model. The nonparametric model is a Gaussian process model, and Logistic regression is the parametric counterpart. Since our fairness approach is not being framed as a constrained optimisation problem, we can reuse off-the-shelf toolboxes including the GPyTorch library by Gardner et al. (2018) for Gaussian process models. This library incorporates recent advances in scalable variational inference including variational *inducing inputs* and likelihood ratio/REINFORCE estimators. The variational posterior can be derived from the likelihood and the prior. We need just need to modify the likelihood to take into account the target labels (Algorithm 1).

4.6.2 *Data*

We run experiments on two real-world datasets. The first dataset is the ADULT INCOME dataset (Dheeru and Karra Taniskidou, 2017). It contains 33,561 data points with census information from US citizens. The labels indicate whether the individual earns more ($y = 1$) or less ($y = 0$) than \$50,000 per year. We use the dataset with either *race* or *gender* as the sensitive attribute. The input dimension, excluding the sensitive attributes, is 12 in the raw data; the categorical features are then one-hot encoded. For the experiments, we removed 2,399 instances with missing data and used only the training data, which we split randomly for each trial run. The second dataset is the PROPUBLICA RECIDIVISM dataset. It contains data from 6,167 individuals that were arrested. The data was collected when investigating the COMPAS risk assessment tool (Angwin et al., 2016). The task is to predict whether the person was rearrested within two years ($y = 1$ if they were rearrested, $y = 0$ otherwise). We again use the dataset with either *race* or *gender* as the sensitive attributes.

4.6.3 *Balancing the test set*

Any fairness method that is targeting demographic parity, treats the training set as defective in one way: the acceptance rates are not equal in the training set and this needs to be corrected. As such, it does not make sense to evaluate these methods on a dataset that is equally defective. Predicting at equal acceptance rates is the correct result and the test set should reflect this.

In order to generate a test set which has the property of equal acceptance rates, we subsample the given, imbalanced, test set. For evaluating demographic parity, we discard datapoints from the imbalanced test set such that the resulting subset satisfies $P(s = j|y = i) = \frac{1}{2}$ for all i and j. This balances the set in terms of s and ensures $P(y, s) = P(y)P(s)$, but does not force the acceptance rate to be $\frac{1}{2}$, which in the case of the Adult dataset would be a severe change as the acceptance rate is naturally quite low there. Using the described method ensures that the minimal amount of data is discarded for the Adult dataset. We have empirically observed that all fairness algorithms benefit from this balancing of the test set.

The situation is different for equality of opportunity. A perfect classifier automatically satisfies equality of opportunity on *any dataset*. Thus, an algorithm aiming for this fairness constraint should not treat the dataset as defective. Consequently, for evaluating equality of opportunity we perform no balancing of the test set.

Table 4.1: Accuracy and fairness (with respect to *demographic parity*) for various methods on the balanced test set of the Adult dataset. Fairness is defined as $PR_{s=0}/PR_{s=1}$ (a completely fair model would achieve a value of 1.0). Left: using RACE as the sensitive attribute. Right: using GENDER as the sensitive attribute. The mean and std of 10 repeated experiments.

Algorithm	Fair → 1.0 ←	Accuracy ↑	Fair → 1.0 ←	Accuracy ↑
GP	0.80 ± 0.07	0.888 ± 0.007	0.54 ± 0.05	0.900 ± 0.006
LR	0.83 ± 0.06	0.884 ± 0.007	0.52 ± 0.03	0.898 ± 0.003
SVM	0.89 ± 0.06	0.899 ± 0.004	0.49 ± 0.05	0.913 ± 0.004
FairGP (ours)	0.86 ± 0.07	0.888 ± 0.006	0.87 ± 0.09	0.902 ± 0.007
FairLR (ours)	0.90 ± 0.06	0.874 ± 0.009	0.93 ± 0.04	0.886 ± 0.012
ZafarAccuracy	0.67 ± 0.17	0.808 ± 0.016	0.77 ± 0.08	0.853 ± 0.017
ZafarFairness	0.81 ± 0.06	0.879 ± 0.009	0.74 ± 0.11	0.897 ± 0.004
Kamiran and Calders (2012)	0.87 ± 0.07	0.882 ± 0.007	0.96 ± 0.03	0.900 ± 0.004
Agarwal et al. (2018)	0.86 ± 0.08	0.883 ± 0.008	0.65 ± 0.04	0.900 ± 0.004

4.6.4 *Method*

We evaluate two versions of our target label model[1]: *FairGP*, which is based on Gaussian Process models, and *FairLR*, which is based on logistic regression. We also train baseline models that do not take fairness into account.

In both *FairGP* and *FairLR*, our approach is implemented by modifying the likelihood function. First, the unmodified likelihood is computed (corresponding to $P(\tilde{y} = 1|x, \theta)$) and then a linear transformation (dependent on s) is applied as given by equation 4.13. No additional ranking of the samples is needed, because the unmodified likelihood already supplies ranking information.

The fair GP models and the baseline GP model are all based on variational inference and use the same settings. During training, each batch is equivalent to the whole dataset. The number of inducing inputs is 500 on the ProPublica dataset and 2500 on the Adult dataset which corresponds to approximately 1/8 of the number of training points for each dataset. We use a squared-exponential (SE) kernel with automatic relevance determination (ARD) and the probit function as the likelihood function. We optimise the hyper-parameters and the variational parameters using the Adam method (Kingma and Ba, 2015) with the default parameters. We use the full covariance matrix for the Gaussian variational distribution. The logistic regression is trained with RAdam (Liu et al., 2020) and uses L2 regularisation. For the regularisation coefficient, we conducted a hyper-parameter search over 10 folds of the data. For each fold, we picked the hyper-parameter which achieved the best fairness among those 5 with the best accuracy scores. We

1 The code can be found on GitHub: https://github.com/predictive-analytics-lab/ethicml-models/tree/master/implementations/fairgp.

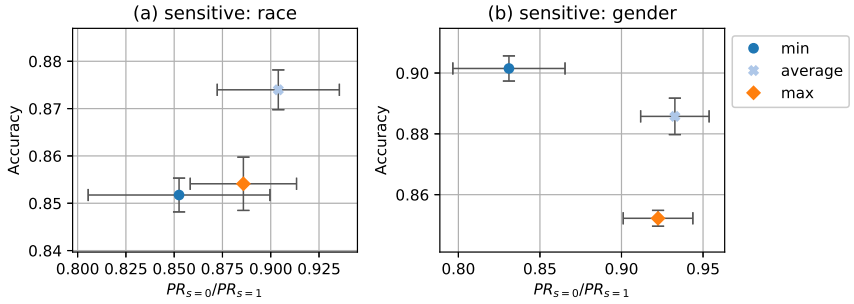


Figure 4.1: Accuracy and fairness (demographic parity) for various target choices. (a): Adult dataset using race as the sensitive attribute; (b): Adult dataset using gender. Centre of the cross is the mean; height and width of the box encode half of standard derivation of accuracy and disparate impact.

then averaged over the 10 hyper-parameter values chosen in this way and then used this average for all runs to obtain our final results.

In addition to the GP and LR baselines, we compare our proposed model with the following methods: Support Vector Machine (*SVM*), *Kamiran & Calders* (Kamiran and Calders, 2012) ("reweighing" method), *Agarwal et al.* (Agarwal et al., 2018) (using logistic regression as the classifier) and several methods given by Zafar et al. (2017a,b), which include maximising accuracy under demographic parity fairness constraints (*ZafarFairness*), maximising demographic parity fairness under accuracy constraints (*ZafarAccuracy*), and removing disparate mistreatment by constraining the false negative rate (*ZafarEqOpp*). Every method is evaluated over 10 repeats that each have different splits of the training and test set.

4.6.5 *Results for Demographic Parity on Adult dataset*

Following Zafar et al. (2017b) we evaluate demographic parity on the Adult dataset. Table 4.1 shows the accuracy and fairness for several algorithms. In the table, and in the following, we use $PR_{s=i}$ to denote the observed rate of positive predictions per demographic group $P(\hat{y} = 1|s = i)$. Thus, $PR_{s=0}/PR_{s=1}$ is a measure for demographic parity, where a completely fair model would attain a value of 1.0. This measure for demographic parity is also called "disparate impact" (see e.g. Feldman et al., 2015; Zafar et al., 2017a). As the results in Table 4.1 show, FairGP and FairLR are clearly fairer than the baseline GP and LR. We use the mean (PR_t^{avg}) for the target acceptance rate. The difference between fair models and unconstrained models is not as large with *race* as the sensitive attribute, as the unconstrained models are already quite fair there. The results of FairGP are characterised by high fairness and

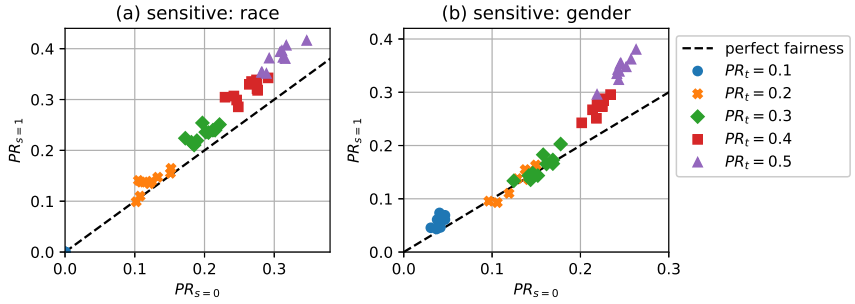


Figure 4.2: Predictions with different target acceptance rates (demographic parity) for 10 repeats. (a): $PR_{s=0}$ vs $PR_{s=1}$ using race as the sensitive attribute; (b): $PR_{s=0}$ vs $PR_{s=1}$ using gender.

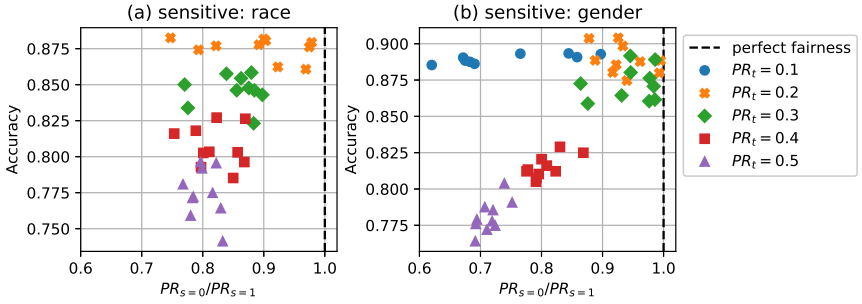


Figure 4.3: Predictions with different target acceptance rates (demographic parity) for 10 repeats. (a): disparate impact vs accuracy on Adult dataset using race as the sensitive attribute; (b): disparate impact vs accuracy using gender.

high accuracy. FairLR achieves similar results to FairGP, but with generally slightly lower accuracy but better fairness. We used the two step procedure of Donini et al. (2018) to verify that we cannot achieve the same fairness result with just parameter search on LR.

In fig. 4.1, we investigate which choice of target (PR_t^{avg}, PR_t^{min} or PR_t^{max}) gives the best result. We use PR_t^{avg} for all following experiments as this is the fairest choice (cf. section 4.4.2). The fig. 4.1(a) shows results from Adult dataset with *race* as sensitive attribute where we have $PR_t^{min} = 0.156$, $PR_t^{max} = 0.267$ and $PR_t^{avg} = 0.211$. PR_t^{avg} performs best in term of the trade-off.

Fig. 4.2(a) and (b) show runs of FairLR where we explicitly set a target acceptance rate, $PR_t := P(\bar{y} = 1)$, instead of taking the mean PR_t^{avg}. A perfect targeting mechanism would produce a diagonal. The plot shows that setting the target rate has the expected effect on the observed acceptance rate. This tuning of the target rate is the unique aspect of the approach. This would be very difficult to achieve with existing fairness methods; a new constraint would have to be added. The achieved positive rate is, however, usually a

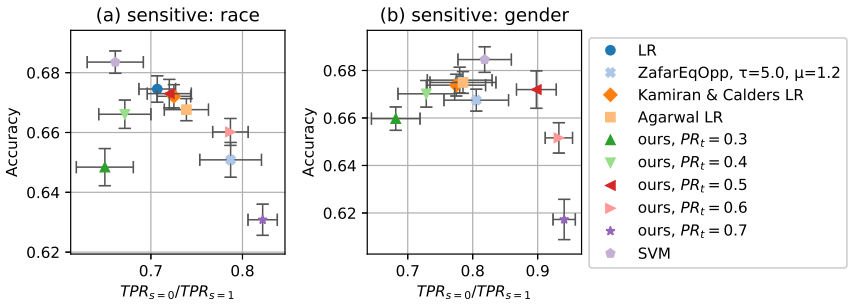


Figure 4.4: Accuracy and fairness (with respect to *equality of opportunity*) for various methods on ProPublica dataset. (a): using race as the sensitive attribute; (b): using gender. A completely fair model would achieve a value of 1.0 in the x-axis. See fig. 4.5(a) and (b) on how these choices of PR setting translate to $TPR_{s=0}$ vs $TPR_{s=1}$.

bit lower than the targeted rate (e.g. around 0.15 for the target 0.2). This is due to using imperfect classifiers; if TPR and TNR differ from 1, the overall positive rate is affected (see e.g. Forman (2005) for discussion of this).

Fig. 4.3(a) and (b) show the same data as fig. 4.2 but with different axes. It can be seen from this fig. 4.3(a) and (b) that the fairness-accuracy trade-off is usually best when the target rate is close to the average of the positive rates in the dataset (which is around 0.2 for both sensitive attribute).

4.6.6 *Results for Equality of Opportunity on ProPublica dataset.*

For equality of opportunity, we again follow Zafar et al. (2017a) and evaluate the algorithm on the ProPublica dataset. As we did for demographic parity, we define a measure of equality of opportunity via the ratio of the true positive rates (TPRs) within the demographic groups. We use $TPR_{s=i}$ to denote the observed TPR in group i: $P(\hat{y} = 1|y = 1, s = i)$, and $TNR_{s=i}$ for the observed true negative rate (TNR) in the same manner. The measure is then given by $TPR_{s=0}/TPR_{s=1}$. A perfectly fair algorithm would achieve 1.0 on the measure.

The results of 10 runs are shown in fig. 4.4 and fig. 4.5. Fig. 4.4(a) and (b) show the accuracy-fairness trade-off; fig. 4.5(a) and (b) show the achieved TPRs. In the accuracy-fairness plot, varying PR_t is shown to produce an inverted U-shape: Higher PR_t still leads to improved fairness, but at a high cost in terms of accuracy.

The latter two plots make clear that the TPR ratio does not tell the whole story: the realisation of the fairness constraint can differ substantially. By setting different target PRs for our method, we can affect TPRs as well, where higher PR_t leads to higher TPR, stemming from the fact that making more positive predictions increases the chance of making correct positive

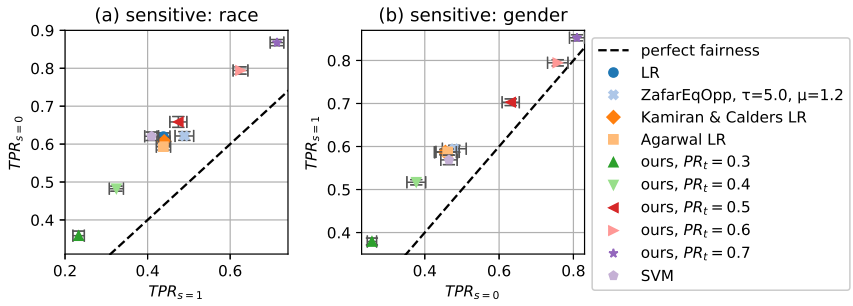


Figure 4.5: Fairness measure $TPR_{s=0}$ vs $TPR_{s=1}$ (*equality of opportunity*) for different target PRs (PR_t). (a): on dataset ProPublica recidivism using race as the sensitive attribute; (b): using gender.

predictions. Fig. 4.5 shows that our method can span a wide range of possible TPR values. Tuning these hidden aspects of fairness is the strength of our method.

4.7 Discussion and conclusion

Fairness is fundamentally not a challenge of algorithms alone, but very much a sociological challenge. A lot of proposals have emerged recently for defining and obtaining fairness in machine learning-based decision making systems. The vast majority of academic work has focused on two categories of definitions: statistical (group) notions of fairness and individual notions of fairness (see Verma and Rubin (2018) for at least twenty different notions of fairness). Statistical notions are easy to verify but do not provide protections to individuals. Individual notions do give individual protections but need strong assumptions, such as the availability of an agreed-upon similarity metric, which can be difficult in practice. We acknowledge that a proper solution to algorithmic fairness cannot rely on statistics alone. Nevertheless, these statistical fairness definitions can be helpful in understanding the problem and working towards solutions. To facilitate this, at every step, the trade-offs that are present should be made very clear and long-term effects have to be considered as well (Kallus and Zhou, 2018; Liu et al., 2019).

Here, we have developed a machine learning framework which allows us to learn from an implicit balanced dataset, thus satisfying the two most popular notions of fairness (Verma and Rubin, 2018), demographic parity (also known as *avoiding disparate treatment*) and equality of opportunity (or *avoiding disparate mistreatment*). Additionally, we indicate how to extend the framework to cover conditional demographic parity as well. The framework allows us to set a *target rate* to control how the fairness constraint is realised. For example, we can set the target positive rate for demographic parity to be

0.6 for different groups. Depending on the application, it can be important to specify whether non-discrimination ought to be achieved by more positive predictions or more negative predictions. This capability is unique to our approach and can be used as an intuitive mechanism to control the realisation of fairness. Our framework is general and will be applicable for sensitive variables with binary and multi-level values. The current work focuses on a single binary sensitive variable. Future work could extend our tuning approach to other fairness concepts like the closely related predictive parity group fairness (Chouldechova, 2017) or individual fairness (Dwork et al., 2012).

Furthermore, as currently formulated, the approach is only compatible with likelihood-based models (which covers neural networks). However, the only real requirement is reasonably-well calibrated probabilities from the used model, which are needed to compute the expected target labels. Thus, future work could extend it to other kinds of models, like SVMs, for which solutions exists to transform the output to a probability; the most common being Platt scaling (Platt, 1999), which applies a logistic transformation with two free parameters to the output of the SVM. The free parameters have to be estimated by iterating over the whole training set. The remaining (substantial) challenge is then to integrate the loss computed from these probabilities into the training of the SVM.

Acknowledgements

Supported by the UK EPSRC project EP/P03442X/1 'EthicalML: Injecting Ethical and Legal Constraints into Machine Learning Models' and the Russian Academic Excellence Project '5–100'. We gratefully acknowledge NVIDIA for GPU donations, and Amazon for AWS Cloud Credits. We thank Chao Chen and Songzhu Zheng for their inspiration of our main proof.

4.8 Appendix

4.8.1 *Proof of Theorem 1*

Let $\eta(x,s) = P(y = 1|x,s)$ be the distribution of the training data. Let $\bar{\eta}(x,s) = m_s \cdot \eta(x,s) + b_s$, where

$$\begin{aligned} m_s &= P(\bar{y} = 1|y = 1, s) - P(\bar{y} = 1|y = 0, s) \\ &= 1 - P(\bar{y} = 0|y = 1, s) - P(\bar{y} = 1|y = 0, s) \qquad (4.32) \\ b_s &= P(\bar{y} = 1|y = 0, s) \qquad (4.33) \end{aligned}$$

So, $\bar{\eta}(x,s) = P(\bar{y} = 1|x,s)$. Let y denote the *hard* labels for η: $y = \mathbb{I}\left[\eta > \frac{1}{2}\right]$ and $\bar{y}$ be the hard labels for $\bar{\eta}$: $\bar{y} = \mathbb{I}\left[\bar{\eta} > \frac{1}{2}\right]$.

THEOREM 4.3. *The probability that y and $\bar{y}$ disagree ($y \neq \bar{y}$) for any input x in the dataset is given by:*

$$P(y \neq \bar{y}|s) = P\left(\left|\eta(x,s) - \tfrac{1}{2}\right| < t_s\right) \qquad (4.34)$$

where

$$t_s = \left|\frac{m_s + 2b_s - 1}{2m_s}\right| . \qquad (4.35)$$

Proof. The decision boundary that lets us recover the true labels is at $\frac{1}{2}$ (independent of s). So, for the shifted distribution, $\bar{\eta}$, this threshold to get the true labels would be at $\frac{1}{2} \cdot m_s + b_s$ (it depends on s now). If we however use the decision boundary of $\frac{1}{2}$ for $\bar{\eta}$, to make our predictions, $\bar{y}$, then this prediction will sometimes not correspond to the true label, $y \neq \bar{y}$. When does this happen?

Let d_s be the new decision boundary: $d_s = \frac{1}{2} \cdot m_s + b_s$. There are two possibilities to consider here: either $\frac{1}{2} < d_s$ or $\frac{1}{2} > d_s$ (for $d_s = \frac{1}{2}$, the decision boundaries are the same and nothing has to be shown). The problem, $y \neq \bar{y}$, appears then exactly when the value of $\bar{\eta}$ is between the two boundaries:

$$\begin{aligned} &\text{if } d_s > \tfrac{1}{2}\text{:} \quad d_s > \bar{\eta}(x,s) > \tfrac{1}{2} \qquad (4.36) \\ &\text{if } d_s < \tfrac{1}{2}\text{:} \quad d_s < \bar{\eta}(x,s) < \tfrac{1}{2} \qquad (4.37) \end{aligned}$$

Expressing this in terms of η and simplifying leads to (if m_s is negative, then the two cases are swapped, but we still get both inequalities):

$$\text{if } d_s > \tfrac{1}{2}: \quad \frac{1}{2} > \eta(x,s) > \frac{1-2b_s}{2m_s} \tag{4.38}$$

$$\text{if } d_s < \tfrac{1}{2}: \quad \frac{1}{2} < \eta(x,s) < \frac{1-2b_s}{2m_s} \tag{4.39}$$

This can be summarised as

$$\left|\eta(x,s) - \frac{1}{2}\right| < \left|\frac{1}{2} - \frac{1-2b_s}{2m_s}\right| . \tag{4.40}$$

Let t_s denote the term on the right side of this inequality (i.e. the "threshold" that determines whether $y = \bar{y}$ or not). Then

$$t_s = \left|\frac{1}{2} - \frac{1-2b_s}{2m_s}\right| = \left|\frac{m_s + 2b_s - 1}{2m_s}\right| . \tag{4.41}$$

So, we have: $\left|\eta(x,s) - \frac{1}{2}\right| < t_s = \left|\frac{m_s+2b_s-1}{2m_s}\right|$. This leads directly to the statement we wanted to prove:

$$P(y \neq \bar{y}|s) = P\left(\left|\eta(x,s) - \frac{1}{2}\right| < t_s\right) . \tag{4.42}$$

□

4.8.2 *Finding minimal* t_s

We express t_s in terms of PR_b^s and PR_t.

$$t_s = \begin{cases} \frac{1}{2}\frac{PR_b^s - PR_t}{PR_t} & \text{if } PR_t > PR_b^j \\ \frac{1}{2}\frac{PR_t - PR_b^s}{1-PR_t} & \text{otherwise.} \end{cases} \tag{4.43}$$

Without loss of generality, we assume $PR_b^0 < PR_b^1$. As mentioned in the main text, PR_t should be between PR_b^0 and PR_b^1 to minimise both t_s. If that is the case, then we get

$$t_{s=0} = \frac{1}{2}\frac{PR_t - PR_b^0}{1 - PR_t} \tag{4.44}$$

$$t_{s=1} = \frac{1}{2}\frac{PR_b^1 - PR_t}{PR_t} . \tag{4.45}$$

If we further assume $PR_b^1 < \frac{1}{2}$, then we also have $PR_t < \frac{1}{2}$ and thus $PR_t < 1 - PR_t$. This implies that the denominator of $t_{s=1}$ is smaller and that, in turn,

$t_{s=1}$ grows faster. This faster growth means that when minimising $t_{s=0} + t_{s=1}$, we have to concentrate on $t_{s=1}$. The minimum is then such that $t_{s=1}$ is 0, i.e. $PR_t = PR_b^1$.

4.8.3 *Proof of Theorem 2*

We are given a dataset $\mathscr{D} = \{(x_i, y_i)\}_i$, where the x_i are vectors of features and the y_i the corresponding labels. We refer to the tuples (x, y) as the *samples* of the dataset. The number of samples is $N = |\mathscr{D}|$.

We assume binary labels ($y \in \{0, 1\}$) and thus can form the (disjoint) subsets $\mathscr{Y}^0$ and $\mathscr{Y}^1$ with

$$\mathscr{Y}^j = \{(x, y) \in \mathscr{D} | y = j\} \quad \text{with } j \in \{0, 1\}\,. \tag{4.46}$$

Furthermore, we associate each sample with a classification $\hat{y} \in \{0, 1\}$. The task of making the classification $\hat{y} = 0$ or $\hat{y} = 1$ can be understood as putting each sample from $\mathscr{D}$ into one of two sets: $\mathscr{C}^0$ and $\mathscr{C}^1$, such that $\mathscr{C}^0 \cup \mathscr{C}^1 = \mathscr{D}$ and $\mathscr{C}^0 \cap \mathscr{C}^1 = \emptyset$.

We refer to the set $\mathscr{A} = (\mathscr{C}^0 \cap \mathscr{Y}^0) \cup (\mathscr{C}^1 \cap \mathscr{Y}^1)$ as the set of correct (or *a*ccurate) predictions. The *accuracy* is given by $acc = N^{-1} \cdot |\mathscr{A}|$. From the definition it is clear that $0 \leq acc \leq 1$.

Definition 4.4.

$$r_a := \frac{|\mathscr{Y}^1|}{|\mathscr{D}|} = \frac{|\mathscr{Y}^1|}{N} \tag{4.47}$$

is called the *acceptance rate* of the dataset $\mathscr{D}$.

Definition 4.5.

$$\hat{r}_a = \frac{|\mathscr{C}^1|}{|\mathscr{D}|} = \frac{|\mathscr{C}^1|}{N} \tag{4.48}$$

is called the *target rate* of the predictions.

Theorem 4.4. *For a dataset with the acceptance rate r_a and corresponding predictions with a target rate of $\hat{r}_a$, the accuracy is limited by*

$$acc \leq 1 - |\hat{r}_a - r_a|\ . \tag{4.49}$$

Proof. We first note that by multiplying by N, the inequality becomes

$$|\mathscr{A}| \leq N - \big||\mathscr{C}^1| - |\mathscr{Y}^1|\big|\ . \tag{4.50}$$

We will choose the predictions $\hat{y}$ that achieve the highest possible accuracy (largest possible $\mathscr{A}$) and show that this can never exceed $1 - |\hat{r}_a - r_a|$. As the set $\mathscr{Y}^1$ contains all samples that correspond to $y = 1$, we try to take as many samples from $\mathscr{Y}^1$ for $\mathscr{C}^1$ as possible. Likewise, we take as many indices as possible from $\mathscr{Y}^0$ for $\mathscr{C}^0$.

We consider three cases: $\hat{r}_a = r_a$, $\hat{r}_a < r_a$ and $\hat{r}_a > r_a$. The first case is trivial; we have $|\mathscr{C}^1| = |\mathscr{Y}^1|$ and thus are able to set $\mathscr{C}^1 = \mathscr{Y}^1$, $\mathscr{C}^0 = \mathscr{Y}^0$ and achieve perfect accuracy ($acc \leq 1$).

For $\hat{r}_a < r_a$, we have $|\mathscr{C}^1| < |\mathscr{Y}^1|$ and thus have more samples available with $y = 1$ than we would optimally need to select for $\mathscr{C}^1$. There are two terms to consider that make up the definition of $\mathscr{A}$: $\mathscr{C}^0 \cap \mathscr{Y}^0$ and $\mathscr{C}^1 \cap \mathscr{Y}^1$. The intersection of these two terms is empty because $\mathscr{C}^0 \cap \mathscr{C}^1 = \emptyset$. Thus,

$$|\mathscr{A}| = \left|(\mathscr{C}^0 \cap \mathscr{Y}^0) \cup (\mathscr{C}^1 \cap \mathscr{Y}^1)\right| = \left|(\mathscr{C}^0 \cap \mathscr{Y}^0)\right| + \left|(\mathscr{C}^1 \cap \mathscr{Y}^1)\right| \ . \tag{4.51}$$

Selecting samples from $\mathscr{Y}^1$ for $\mathscr{C}^0$ will only *decrease* the first term, so for maximum accuracy, it is fine to take as many samples from $\mathscr{Y}^1$ for $\mathscr{C}^1$. Taking all available samples from $\mathscr{Y}^1$ such that $\mathscr{C}^1 \supset \mathscr{Y}^1$, there is still space left in $\mathscr{C}^1$ which we will have to fill with samples with $y = 0$. Thus, we have $\mathscr{C}^1 \cap \mathscr{Y}^1 = \mathscr{Y}^1$. For $\mathscr{C}^0$, we have enough $y = 0$ such that $\mathscr{C}^0 \subset \mathscr{Y}^0$ and $\mathscr{C}^0 \cap \mathscr{Y}^0 = \mathscr{C}^0$. This is the largest we can make these intersections. Putting everything together:

$$\begin{aligned} \left|\mathscr{A}^{optimal}\right| &= \left|(\mathscr{C}^0 \cap \mathscr{Y}^0)\right| + \left|(\mathscr{C}^1 \cap \mathscr{Y}^1)\right| = \left|\mathscr{C}^0\right| + \left|\mathscr{Y}^1\right| \\ &= N - \left|\mathscr{C}^1\right| + \left|\mathscr{Y}^1\right| = N - \left(\left|\mathscr{C}^1\right| - \left|\mathscr{Y}^1\right|\right) \ . \end{aligned} \tag{4.52}$$

For $\hat{r}_a > r_a$, the roles of $\mathscr{C}^0$ and $\mathscr{C}^1$ are reversed and thus, the signs in the equation are inverted:

$$\left|\mathscr{A}^{optimal}\right| = N - (\left|\mathscr{Y}^1\right| - \left|\mathscr{C}^1\right|) \ . \tag{4.53}$$

This proves the claim. □

Corollary 4.4.1. *Given a dataset that consists of two subsets $\mathcal{S}_0$ and $\mathcal{S}_1$ ($\mathscr{D} = \mathcal{S}_0 \cup \mathcal{S}_1$) where p is the ratio of $|\mathcal{S}_0|$ to $|\mathscr{D}|$ and given corresponding acceptance rates r_a^0 and r_a^1 and predictions with target rates $\hat{r}_a^0$ and $\hat{r}_a^1$, the accuracy is limited by*

$$acc \leq 1 - p \cdot \left|\hat{r}_a^0 - r_a^0\right| - (1 - p) \cdot \left|\hat{r}_a^1 - r_a^1\right| \ . \tag{4.54}$$

Example 4.1. We consider the case where S_0 (which could for example be all data points for female individuals) makes up 30% of the dataset; so $p = 0.3$.

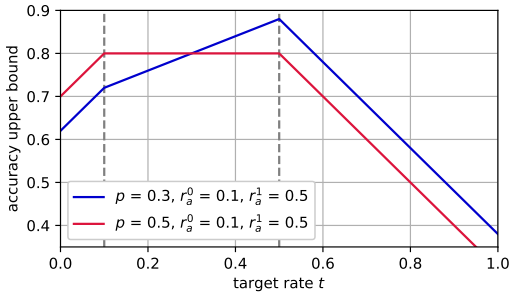


Figure 4.6: Achievable accuracy for different target values.

Further, we say that for S_0 we have an acceptance rate of 10% ($r_a^0 = 0.1$) and for S_1, 50% ($r_a^1 = 0.5$). If we then set both target rates to the same value t ($\hat{r}_a^0 = \hat{r}_a^1 = t$), with $t = 0.3$, then the highest accuracy that can be achieved is 0.8 or 80%.

Fig 4.6 shows the achievable accuracy for different values of t in blue: We can see that we can achieve the highest accuracy for $t = r_a^1 = 0.5$, namely 88%. The plot in orange shows the achievable accuracy for $p = 0.5$, i.e., when the two subsets have the same size. In this case, all target rates between r_a^0 and r_a^1 give equal results, namely 80%.

4.8.4 *Illustration of restrictions on PR*

We start by setting a target rate r_t:

$$P(\bar{y} = 1|s = 0) \stackrel{!}{=} r_t \quad \text{and} \quad P(\bar{y} = 1|s = 1) \stackrel{!}{=} r_t \tag{4.55}$$

This leads us to the following constraint for $s' \in \{0, 1\}$:

$$r_t = P(\bar{y} = 1|s = s') = \sum_y P(\bar{y} = 1|y, s = s')P(y|s = s') \tag{4.56}$$

For $P(y|s = s')$ we will put in the value at which we want our constraint to hold. We use r_b^j to denote the base rate $P(y = 1|s = j)$ which we estimate from the training set. Plugging this in, we are left with

$$r_t = P(\bar{y} = 1|y = 0, s = 0) \cdot (1 - r_b^0) + P(\bar{y} = 1|y = 1, s = 0) \cdot r_b^0 \tag{4.57}$$

$$r_t = P(\bar{y} = 1|y = 0, s = 1) \cdot (1 - r_b^1) + P(\bar{y} = 1|y = 1, s = 1) \cdot r_b^1 \,. \tag{4.58}$$

This is a system of linear equations with two equations and four free variables. There is thus still considerable freedom in how we want our constraint to be realised. The freedom that we have here concerns how strongly the accuracy will be affected.

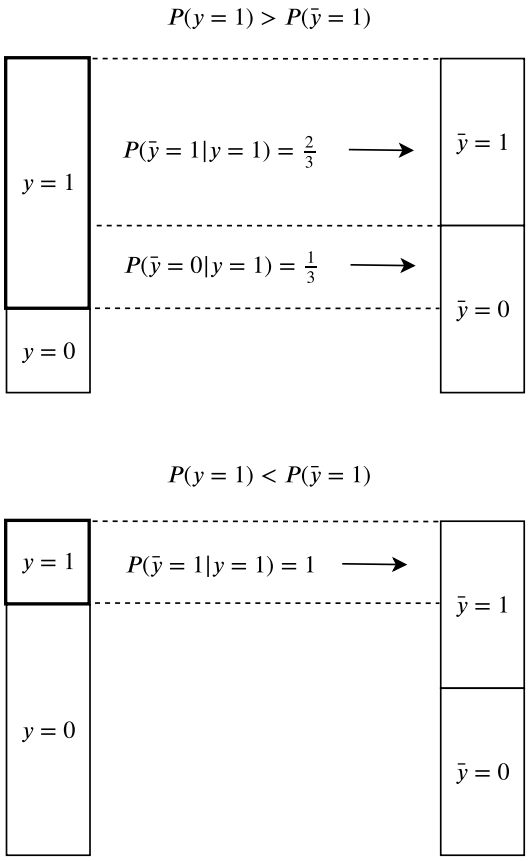


Figure 4.7: Illustration of demographic parity with target labels. In the situation in the upper part, $P(\bar{y} = 1|y = 1)$ cannot be set to 1, because there are more samples with $y = 1$ than there are $\bar{y} = 1$. In the situation in the lower part, $P(\bar{y} = 1|y = 1)$ can be set to 1.

If we set $P(\bar{y} = 1|y = 1, s)$ to 0.5, then we express the fact that a train label of 1 only implies a target label of 1 in 50% of the cases. In order to minimise the effect on accuracy, we make $P(\bar{y} = 1|y = 1, s)$ as high as possible and $P(\bar{y} = 1|y = 0, s)$ as low as possible.

We solve for $P(\bar{y} = 1|y = 0, s = j)$:

$$P(\bar{y} = 1|y = 0, s = j) = \frac{r_b^j}{1 - r_b^j} \left(\frac{r_t}{r_b^j} - P(\bar{y} = 1|y = 1, s = j) \right) . \tag{4.59}$$

However, we can set $P(\bar{y} = 1|y = 0, s = j)$ to 0 only if that does not imply $P(\bar{y} = 1|y = 1, s = j)$ will be greater than 1. This would happen if r_t/r_b^j were greater than 1.

Figure 4.7 illustrates this. In the upper part of the figure, we have r_t/r_b^j less than 1. This means, the target positive rate is less than the base positive rate, and implies that the positive rate has to be lowered somehow. This is accomplished by mapping some of the $y = 1$ samples to $\bar{y} = 0$. Thus,

$P(\bar{y} = 1|y = 1, s = j)$ is less than 1, and $P(\bar{y} = 0|y = 1, s = j)$ greater than 0. In the lower part of the figure, we have the opposite case; essentially, $\bar{y}$ and y swap places. Here, $P(\bar{y} = 1|y = 1, s = j) = 1$ is possible.

5 PAPER 2: NULL-SAMPLING FOR INTERPRETABLE AND FAIR REPRESENTATIONS

Authors:
Thomas Kehrenberg[1], Myles Bartlett[1], Oliver Thomas[1], and Novi Quadrianto[1]
Affiliations:
[1] Predictive Analytics Lab (PAL), University of Sussex, Brighton, UK
Conference: *European Conference on Computer Vision* (ECCV), 2020
DOI: 10.1007/978-3-030-58574-7_34
Note: The appendix has been included as section 5.7.

5.1 Abstract

We propose to learn invariant representations, in the data domain, to achieve interpretability in algorithmic fairness. Invariance implies a selectivity for high level, relevant correlations w.r.t. class label annotations, and a robustness to irrelevant correlations with protected characteristics such as race or gender. We introduce a non-trivial setup in which the training set exhibits a strong bias such that class label annotations are irrelevant and spurious correlations cannot be distinguished. To address this problem, we introduce an adversarially trained model with a *null-sampling* procedure to produce invariant representations in the data domain. To enable disentanglement, a partially-labelled *representative* set is used. By placing the representations into the data domain, the changes made by the model are easily examinable by human auditors. We show the effectiveness of our method on both image and tabular datasets: Coloured MNIST, the CelebA and the Adult dataset.

5.2 Introduction

Without due consideration for the data collection process, machine learning algorithms can exacerbate biases, or even introduce new ones if proper control is not exerted over their learning (Holstein et al., 2019). While most of these issues can be solved by controlling and curating data collection in a fairness-conscious fashion, doing so is not always an option, such as when working with historical data. Efforts to address this problem algorithmically have been centred on developing statistical definitions of fairness and learning

models that satisfy these definitions. One popular definition of fairness used to guide the training of fair classifiers, for example, is *demographic parity*, stating that positive outcome rates should be equalised (or *invariant*) across protected groups.

In the typical setup, we have an input $\boldsymbol{x}$, a sensitive attribute s that represents some non-admissible information like gender and a class label y which is the prediction target. The idea of fair *representation* learning (Zemel et al., 2013; Edwards and Storkey, 2016; Madras et al., 2018) is then to transform the input $\boldsymbol{x}$ to a representation $\boldsymbol{z}$ which is invariant to s. Thus, learning from $\boldsymbol{z}$ will not introduce a forbidden dependence on s. A good fair representation is one that preserves most of the information from $\boldsymbol{x}$ while satisfying the aforementioned constraints.

As unlabelled data is much more freely available than labelled data, it is of interest to learn the representation in an unsupervised manner. This will allow us to draw on a much more diverse pool of data to learn from. While annotations for y are often hard to come by (and often noisy; see Kehrenberg et al., 2020a), annotations for the sensitive attribute s are usually less so, as s can often be obtained from demographic information provided by census data. We thus consider the setting where the representation is learned from data that is only labelled with s and not y. This is in contrast to most other representation learning methods. We call the set used to learn the representation the *representative* set, because its distribution is meant to match the distribution of the deployment setting (and is thus representative).

Once we have learnt the mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$, we can transform the *training* set which, in contrast to the representative set, has the y labels (and s labels). In order to make our method more widely applicable, we consider an *aggravated fairness problem* in which the training set contains a strong spurious correlation between s and y, which makes it impossible to learn from it a representation which is invariant to s but not invariant to y. Non-invariance to y is important in order to be able to predict y. The training set thus does *not* match the deployment setting, thereby rendering the representative set essential for learning the right invariance. From hereon, we will use the terms *spurious* and *sensitive* interchangeably, depending on the context, to refer to an attribute of the data we seek invariance to. We can draw a connection between learning in the presence of spurious correlations and what Kallus and Zhou (2018) call *residual unfairness*. Consider the Stop, Question and Frisk (SQF) dataset for example: the data was collected in New York City, but the demographics of the recorded cases do not represent the true demographics of NYC well. The demographic attributes of the recorded individuals might correlate so strongly with the prediction target that the two

are nearly indistinguishable. This is the scenario that we are investigating: s and y are so closely correlated in the labelled dataset that they cannot be distinguished, but the learning of s is favoured due to being the "path of least resistance". The deployment setting (i.e. the test set) does not possess this strong correlation and thus a naïve approach will lead to very unfair predictions. In this case, a disentangled representation is insufficient; the representation needs to be explicitly invariant solely with respect to s. In our approach, we make use of the (partially labelled) representative set to learn this invariant representation.

While there is a substantial body of literature devoted to the problems of fair representation-learning, exactly how the invariance in question is achieved is often overlooked. When critical decisions, such as who should receive bail or be released from jail, are being deferred to an automated decision making system, it is critical that people be able to trust the logic of the model underlying it, whether it be via semantic or visual explanations. We build on the work of Quadrianto et al. (2019) and learn a decomposition ($f^{-1} : Z_s \times Z_{\neg s} \to X$) of the *data domain* ($X$) into independent subspaces *invariant* to s ($Z_{\neg s}$) and *indicative* of s (Z_s), which lends an interpretability that is absent from most representation-learning methods. While model interpretability has no strict definition (Zhang and Zhu, 2018), we follow the intuition of Adel et al. (2018) – *a simple relationship to something we can understand*, a definition which representations in the data domain naturally fulfil.

Whether as a result of the aforementioned sampling bias or simply because the features necessarily co-occur, it is not rare for features to correlate with one another in real-world datasets. Lipstick and gender for example, are two attributes that we expect to be highly correlated and to enforce invariance to gender can implicitly enforce invariance to makeup. This is arguably the desired behaviour. However, unforeseen biases in the data may engender cases which are less justifiable. By baking interpretability into our model (by having representations in the data domain), though we still have no better control over what is learned, we can at least diagnose such pathologies.

To render our representations interpretable, we rely on a simple transformation we call *null-sampling* to map invariant representations in the data domain. Previous approaches to fair representation learning (Edwards and Storkey, 2016; Louizos et al., 2016; Beutel et al., 2017; Madras et al., 2018) predominantly rely upon autoencoder models to jointly minimise reconstruction loss and invariance. We discuss first how this can be done with such a model that we refer to as cVAE (conditional VAE), before arguing that the bijectivity of invertible neural networks (INNs) (Dinh et al., 2014)

makes them better suited to this task. We refer to the variant of our method based on these as cFlow (conditional Flow). INNs have several properties that make them appealing for unsupervised representation learning. The focus of our approach is on creating invariant representations that preserve the non-sensitive information maximally, with only knowledge of s and not of the target y, while at the same time having the ability to easily probe what has been learnt.

Our contribution is thus two-fold: 1) We propose a simple approach to generating representations that are invariant to a feature s, while having the benefit of interpretability that comes with being in the data domain. We call our model *NIFR* (*N*ull-sampling for *I*nterpretable and *F*air *R*epresentations). 2) We explore a setting where the labelled training set suffers from varying levels of sampling bias, demonstrating an approach based on transferring information from a more diverse representative set, with guarantees of the non-spurious information being preserved.

5.3 Background

5.3.1 *Learning fair representations.*

Given a sensitive attribute s (for example, gender or race) and inputs $\boldsymbol{x}$, a fair representation $\boldsymbol{z}$ of $\boldsymbol{x}$ is then one for which $\boldsymbol{z} \perp s$ holds, while ideally also being predictive of the class label y. Zemel et al. (2013) was the first to propose the learning of fair representations which allow for transfer to new classification tasks. More recent methods are often based on variational autoencoders (VAEs) (Kingma and Welling, 2014; Edwards and Storkey, 2016; Louizos et al., 2016; Beutel et al., 2017). The achieved fairness of the representation can be measured with various fairness metrics. These measure, however, usually how fair the predictions of a classifier are and not how fair a representation is.

The appropriate measure of fairness for a given task is domain-specific (Liu et al., 2019) and there is often not a universally accepted measure. However, *Demographic Parity* is the most widely used (Edwards and Storkey, 2016; Louizos et al., 2016; Beutel et al., 2017). Demographic Parity demands $\hat{y} \perp s$ where $\hat{y}$ refers to the predictions of the classifier. In the context of fair representations, we measure the Demographic Parity of a downstream classifier, $f(\cdot)$, which is trained on the representation z, i.e. $f : Z \to \hat{Y}$.

A core principle of all fairness methods is the *accuracy-fairness trade-off*. As previously stated, the fair representation should be invariant to s ($\to$

fairness) but still be predictive of y ($\rightarrow$ accuracy). These desiderata cannot, in general, be simultaneously satisfied if s and y are correlated.

The majority of existing methods for fair representations also make use of y labels during training, in order to ensure that $\boldsymbol{z}$ remains predictive of y. This aspect can, in theory, be removed from the methods, but then there is no guarantee that information about y is preserved (Louizos et al., 2016).

5.3.2 *Learning fair, transferrable representations*

In addition to producing fair representations, Madras et al. (2018) want to ensure the representations are transferrable. Here, an adversary is used to remove sensitive information from a representation z. Auxiliary prediction and reconstruction networks, to predict class label y and reconstruct the input x respectively, are trained on top of z, with s being ancillary input to the reconstruction.

Also related is Creager et al. (2019) who employ a FactorVAE (Kim and Mnih, 2018) regularised for fairness. The idea is to learn a representation that is both disentangled and invariant to multiple sensitive attributes. This factorisation makes the latent space easily manipulable such that the different subspaces can be freely removed and composed at test time. Zeroing out the dimensions or replacing them with independent noise imparts invariance to the corresponding sensitive attribute. This method closely resembles ours when we use an invertible encoder. However, the emphasis of our approach is on interpretability, information-preservation, and coping with sampling bias - especially extreme cases where $|\operatorname{supp}(S_{tr} \times Y_{tr})| < |\operatorname{supp}(S_{te} \times Y_{te})|$.

Attempts were made by Quadrianto et al. (2019) prior to this work to learn fair representations in the data domain in order to make it interpretable and transferable. In their work, the input is assumed to be additively decomposable in the feature space into a *fair* and *unfair* component, which together can be used by the decoder to recover the original input. This allows us to examine representations in a human-interpretable space and confirm that the model is not learning a relationship reliant on a sensitive attribute. Though a first step in this direction, we believe such a linear decomposition is not sufficiently expressive to fully capture the relationship between the sensitive and non-sensitive attributes. Our approach allows for the modelling of more complex relationships.

5.3.3 *Learning in the presence of spurious correlations*

Strong spurious correlations make the task of learning a robust classifier challenging: the classifier may learn to exploit correlations unrelated to the true causal relationship between the features and label, and thereby fail to generalise to novel settings. This problem was recently tackled by Kim et al. (2019) who apply a penalty based on the mutual information between the feature embedding and the spurious variable. While the method is effective under mild biasing, we show experimentally that it is not robust to the range of settings we consider.

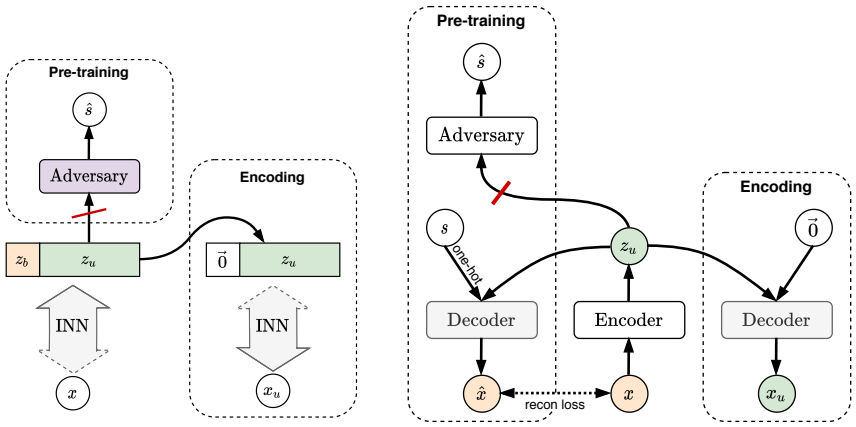
Jacobsen et al. (2019) explore the vulnerability of traditional neural networks to spurious variables – e.g., textures, in the case of ImageNet (Geirhos et al., 2019) – and propose a INN-based solution akin to ours. The INN's encoding is split such that one partition, z_b is encouraged to be predictive of the spurious variable while the other serves as the logits for classification of the semantic label. Information related to the nuisance variable is "pulled out" of the logits as a result of maximising $\log p(s|z_n)$. This specific approach, however, is incompatible with the settings we consider, due to its requirement that both s and y be available at training time.

Viewing the problem from a causal perspective, Arjovsky et al. (2019) develop a variant of empirical risk minimisation called invariant risk minimisation (IRM). The goal of IRM is to train a predictor that generalises across a large set of unseen environments; because variables with spurious correlations do not represent a stable causal mechanism, the predictor learns to be invariant to them. IRM assumes that the training data is not *iid* but is partitioned into distinct environments, $e \in E$. The optimal predictor is then defined as the minimiser of the sum of the empirical risk R_e over this set. In contrast, we assume possession of only a single source of *labelled*, albeit spuriously-correlated, data, but that we have a second source of data that is free of spurious correlations, with the benefit being that it only needs to be labelled *with respect to s*.

5.4 Interpretable Invariances by Null-Sampling

5.4.1 *Problem Statement*

We assume we are given inputs $\boldsymbol{x} \in \mathcal{X}$ and corresponding labels $y \in \mathcal{Y}$. Furthermore, there is some spurious variable $s \in \mathcal{S}$ associated with each input $\boldsymbol{x}$ which we do *not* want to predict. Let X, S and Y be random variables that take on the values $\boldsymbol{x}$, s and y, respectively. The fact that both y and



(a) cFlow model. (b) cVAE model.

Figure 5.1: Training procedure for our models. x: input, s: sensitive attribute, z_u: de-biased representation, x_u: de-biased version of the input in the data domain. The red bar indicates a gradient reversal layer, and $\mathbf{0}$ the null-sampling operation.

s are predictive of $\boldsymbol{x}$ implies that $I(X;Y), I(X;S) > 0$, where $I(\cdot;\cdot)$ is the mutual information. Note, however, that the conditional entropy is non-zero: $H(S|X) \neq 0$, i.e., S is not completely determined by X.

The difficulty of this setup emerges in the training set: there is a close correspondence between S and Y, such that for a model that sees the data through the lens of the loss function, the two are indistinguishable. Furthermore, we assume that this is *not* the case in the test set, meaning the model cannot rely on shortcuts provided by S if it is to generalise from the training set.

We call this scenario where we only have access to the labels of a biasedly-sampled subpopulation an *aggravated fairness problem*. These are not uncommon in the real-world. For instance, in long-feedback systems such as mortgage-approval where the demographics of the subpopulation with observed outcomes is *not* representative of the subpopulation on which the model has been deployed. In this case, s has the potential to act as a false (or *spurious*) indicator of the class label and training a model with such a dataset would limit generalisability. Let (X^{tr}, S^{tr}, Y^{tr}) then be the random variables sampled for the training set and (X^{te}, S^{te}, Y^{te}) be the random variables for the test set. The training and test sets thus induce the following inequality for their mutual information: $I(S^{tr};Y^{tr}) \gg I(S^{te};Y^{te}) \approx 0$.

Our goal is to learn a representation $\boldsymbol{z}_u$ that is independent of s and transferable between downstream tasks. Complementary to $\boldsymbol{z}_u$, we refer to some abstract component of the model that absorbs the unwanted information related to s as $\mathscr{B}$, the realisation of which we define with respect to each of

the two models to be described. The requirement for $\boldsymbol{z}_u$ can be expressed via mutual information:

$$I(\boldsymbol{z}_u; s) \stackrel{!}{=} 0 \,. \tag{5.1}$$

However, for the representation to be useful, we need to capture as much relevant information in the data as possible. Thus, the combined objective function:

$$\min_{\theta} \mathbb{E}_{x\sim X}[-\log p_\theta(\boldsymbol{x})] + \lambda I(f_\theta(x); s) \tag{5.2}$$

where θ refers to the trainable parameters of our model f_θ and $p_\theta(\boldsymbol{x})$ is the likelihood it assigns to the data.

We optimise this loss in an adversarial fashion by playing a min-max game, in which our encoder acts as the generative component. The adversary is an auxiliary classifier g, which receives $\boldsymbol{z}_u$ as input and attempts to predict the spurious variable s. We denote the parameters of the adversary as ϕ; for the parameters of the encoder we use θ, as before. The objective from equation 5.2 is then

$$\min_{\theta\in\Theta} \max_{\phi\in\Phi} \mathbb{E}_{x\sim X}[\log p_\theta(x) - \lambda \mathscr{L}_c(g_\phi(f_\theta(x))); s)] \tag{5.3}$$

where $\mathscr{L}_c$ is the cross-entropy between the predictions for s and the provided labels. In practice, this adversarial term is realised with a gradient reversal layer (GRL) (Ganin et al., 2016) between $\boldsymbol{z}_u$ and g as is common in adversarial approaches (Edwards and Storkey, 2016).

5.4.2 *The Disentanglement Dilemma*

The objective in equation 5.3 balances the two desiderata: predicting y and being invariant to s. However, in the training set (X^{tr}, S^{tr}, Y^{tr}), y and s are so strongly correlated that removing information about s inevitably removes information about y. This strong correlation makes existing methods fail under this setting. In order to even define the right learning goal, we require another source of information that allows us to disentangle s and y. For this, we assume the existence of another set of samples that follow a similar distribution to the test set, but whilst the sensitive attribute is available, the class labels are not. In reality, this is not an unreasonable assumption, as, while properly annotated data is scarce, unlabelled data can be obtained in abundance (with demographic information from census data, electoral rolls, etc.). Previous work has also considered treated "unlabelled data" as still

having s labels (Wick et al., 2019). We are restricted only in the sense that the spurious correlations we want to sever are indicated in the features. We call this the *representative set*, consisting of X^{rep} and S^{rep}. It fulfils $I(S^{rep}; Y^{rep}) \approx 0$ (or rather, it would, if the class labels Y^{rep} were available).

We now summarise the training procedure; an outline for the invertible network model (cFlow) can be seen in fig. 5.1a. First, the encoder network f is trained on (X^{rep}, S^{rep}), during the first phase. The trained network is then used to encode the training set, taking in $\boldsymbol{x}$ and producing the representation, $\boldsymbol{z}_u$, decorrelated from the spurious variable. The encoded dataset can then be used to train any off-the-shelf classifier safely, with information about the spurious variable having been absorbed by some auxiliary component $\mathscr{B}$. In the case of the conditional VAE (cVAE) model, $\mathscr{B}$ takes the form of the decoder subnetwork, which reconstructs the data conditional on a one-hot encoding of s, while for the invertible network $\mathscr{B}$ is realised as a partition of the feature map $\boldsymbol{z}$ (such that $\boldsymbol{z} = [\boldsymbol{z}_u, \boldsymbol{z}_b]$), given the bijective constraint. Thus, the classifier cannot take the shortcut of learning s and instead must learn how to predict y directly. Obtaining the s-invariant representations, $\boldsymbol{x}_u$, in the data domain is simply a matter of replacing the $\mathscr{B}$ component of the decoder's input for the cVAE, and $\boldsymbol{z}_b$ for cFlow, with a zero vector of equivalent size. We refer to this procedure used to generate $\boldsymbol{x}_u$ as *null-sampling* (here, with respect to $\boldsymbol{z}_b$).

Null-sampling resembles the *annihilation* operation described in Xiao et al. (2018), however we note that the two serve very different roles. Whereas the annihilation operation serves as a regulariser to prevent trivial solutions (similar to Jaiswal et al., 2018), null-sampling is used to generate the invariant representations post-training.

5.4.3 *Conditional Decoding*

We first describe a VAE-based model similar to that proposed in Madras et al. (2018), before highlighting some of its shortcomings that motivate the choice of an invertible representation learner.

The model takes the form of a class conditional β-VAE (Higgins et al., 2017), in which the decoder is conditioned on the spurious attribute. We use $\theta_{enc}, \theta_{dec} \in \theta$ to denote the parameters of the encoder and decoder sub-networks, respectively. Concretely, the encoder component performs the mapping $x \to \boldsymbol{z}_u$, while $\mathscr{B}$ is instantiated as the decoder, $\mathscr{B} \coloneqq p_{\theta_{dec}}(x|z_u, s)$, which takes in a concatenation of the learned non-spurious latent vector $\boldsymbol{z}_u$ and a one-hot encoding of the spurious label s to produce a reconstruction of the input $\hat{x}$. Conditioning on a one-hot encoding of s, rather than a single

value, as done in Madras et al. (2018) is the key to visualising invariant representations in the data domain. If $I(z_u; s)$ is properly minimised, the decoder can only derive its information about s from the label, thereby freeing up $\boldsymbol{z}_u$ from encoding the unwanted information while still allowing for reconstruction of the input. Thus, by feeding a zero-vector to the decoder we achieve $\hat{x} \perp s$. The full learning objective for the cVAE is given as

$$\begin{aligned}\mathscr{L}_{\text{cVAE}} =& \mathbb{E}_{q_{\theta_{enc}}(z_u,b|x)}[\log p_{\theta_{dec}}(x|z,b) - \log p_{\theta_{dec}}(s|z_u)] \\ &- \beta D_{KL}(q_{\theta_{enc}}(z_u|x) \| p(z_u))\end{aligned} \tag{5.4}$$

where β is a hyperparameter that determines the trade-off between reconstruction accuracy and independence constraints, and $p(\boldsymbol{z}_u)$ is the prior imposed on the variational posterior. For all our experiments, $p(\boldsymbol{z}_u)$ is realised as an Isotropic Gaussian. Fig. 5.1b summarises the procedure as a diagram.

While we show this setup can indeed work for simple problems, as Madras et al. (2018) before us have, we show that it lacks scalability due to disagreement between the components of the loss. Since information about s is only available to the decoder as a binary encoding, if the relationship between s and x is highly non-linear and cannot be summarised by a simple on/off mechanism, as is the case if s is an attribute such as gender, off-loading information to the decoder by conditioning is no longer possible. As a result, $\boldsymbol{z}_u$ is forced to carry information about s in order to minimise the reconstruction error.

The obvious solution to this is to allow the encoder to store information about s in a partition of the latent space as in Creager et al. (2019). However, we question whether an autoencoder (AE) is the best choice for this setup, with the view that an invertible model is the better tool for the task. Using an invertible model has several guarantees, namely complete information-preservation and freedom from a reconstruction loss, the importance of which we elaborate on below.

5.4.4 *Conditional Flow*

Invertible Neural Networks. Invertible neural networks are a class of neural network architecture characterised by a bijective mapping between their inputs and output (Dinh et al., 2014). The transformations are designed such that their inverses and Jacobians are efficiently computable. These flow-based models permit *exact* likelihood estimation (Rezende and Mohamed, 2015) through the warping of a base density with a series of invertible

transformations and computing the resulting, highly multi-modal, but still normalised, density, using the change of variable theorem:

$$\log p(x) = \log p(z) + \sum \log \left| \det \left(\frac{\mathrm{d}h_i}{h_{i-1}} \right) \right|, \quad p(z) = \mathcal{N}(z; 0, \mathbb{I}) \tag{5.5}$$

where h_i refers to the outputs of the layers of the network and $p(z)$ is the base density, specifically an Isotropic Gaussian in our case. Training of the invertible neural network is then reduced to maximising $\log p(x)$ over the training set, i.e. maximising the probability the network assigns to samples in the training set.

THE BENEFITS OF BIJECTIVITY. Using an invertible network to generate our encoding, $\boldsymbol{z}_u$, carries a number of advantages over other approaches. Ordinarily, the main benefit of flow-based models is that they permit exact density estimation. However, since we are not interested in sampling from the model's distribution, in our case the likelihood term serves as a regulariser, as it does for Jacobsen et al. (2018). Critically, this forces the mean of each latent dimension to zero enabling null-sampling. The invertible property of the network guarantees the preservation of all information relevant to y which is independent of s, regardless of how it is allocated in the output space. Secondly, we conjecture that the encodings are more robust to out-of-distribution data. Whereas an autoencoder (AE) could map a previously seen input and a previously unseen input to the same representation, an invertible network sidesteps this due to the network's bijective property, ensuring all relevant information is stored somewhere. This opens up the possibility of transfer learning between datasets with a similar manifestation of s, as we demonstrate in section 5.7.8.

Under our framework, the invertible network f maps the inputs $\boldsymbol{x}$ to a representation $\boldsymbol{z}_u$: $f(\boldsymbol{x}) = \boldsymbol{z}$. We interpret the embedding $\boldsymbol{z}$ as being the concatenation of two smaller embeddings: $\boldsymbol{z} = [\boldsymbol{z}_u, \boldsymbol{z}_b]$. The dimensionality of $\boldsymbol{z}_b$, and $\boldsymbol{z}_u$, by complement, is a free parameter (see section 5.7.4 for tuning strategies). As f is invertible, $\boldsymbol{x}$ can be recovered like so:

$$\boldsymbol{x} = f^{-1}([\boldsymbol{z}_u, \boldsymbol{z}_b]) \tag{5.6}$$

where $\boldsymbol{z}_b$ is required for equality of the output dimension and input dimension to satisfy the bijectivity of the network – we cannot output $\boldsymbol{z}_u$ alone, but have to output $\boldsymbol{z}_b$ as well. In order to generate the pre-image of $\boldsymbol{z}_u$, we perform null-sampling with respect to $\boldsymbol{z}_b$ by zeroing-out the elements of $\boldsymbol{z}_b$ (such that $\boldsymbol{x}_u = f^{-1}([\boldsymbol{z}_u, \boldsymbol{0}])$), i.e. setting them to the mean of the prior density, $\mathcal{N}(z; 0, I)$.

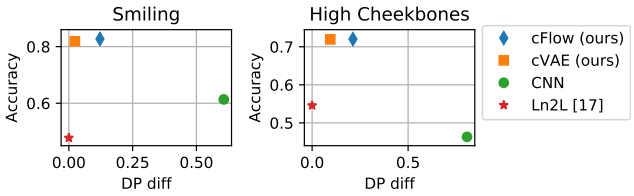


Figure 5.2: Performance of our model for different targets (mixing factor $\eta = 0$). Left: *Smiling* as target, right: *high cheekbones*. *DP diff* measures fairness with respect to demographic parity. A perfectly fair model has a *DP diff* of 0.

How can we be sure that $\boldsymbol{z}_u$ contains enough information about y? The importance of the invertible architecture bears out from this consideration. As long as $\boldsymbol{z}_b$ does not contain the information about y, $\boldsymbol{z}_u$ necessarily must. We can raise or lower the information capacity of $\boldsymbol{z}_b$ by adjusting its size; this should be set to the smallest size sufficient to capture all information about s, so as not to sacrifice class-relevant information. Section 5.7.3 explores the effects of the size further.

5.5 Experiments

We present experiments to demonstrate that the null-sampled representations are in fact invariant to s while still allowing a classifier to predict y from them. We run our cVAE and cFlow models on the coloured MNIST (cMNIST) and CelebA dataset, which we artificially bias, first describing the sampling procedure we follow to do so for non-synthetic datasets. As baselines we have the model of Kim et al. (2019) (Ln2L) and the same CNN used to evaluate the cFlow and cVAE models but with the unmodified images as input (CNN). For the cFlow model we adopt a Glow-like architecture (Kingma and Dhariwal, 2018), while both subnetworks of the cVAE model comprise gated convolutions (Oord et al., 2016), where the encoding size is 256. For cMNIST, we construct the Ln2L baseline according to its original description, for CelebA, we treat it as an augmentation of the baseline CNN's objective function. Detailed information regarding model architectures can be found in sections 5.7.1 and 5.7.4.[1]

5.5.1 *Synthesising Dataset Bias*

For our experiments, we require a training set that exhibits a strong spurious correlation, together with a test set that does not. For cMNIST, this is easily satisfied as we have complete control over the data generation process. For

1 The code can be found at https://github.com/predictive-analytics-lab/nifr.

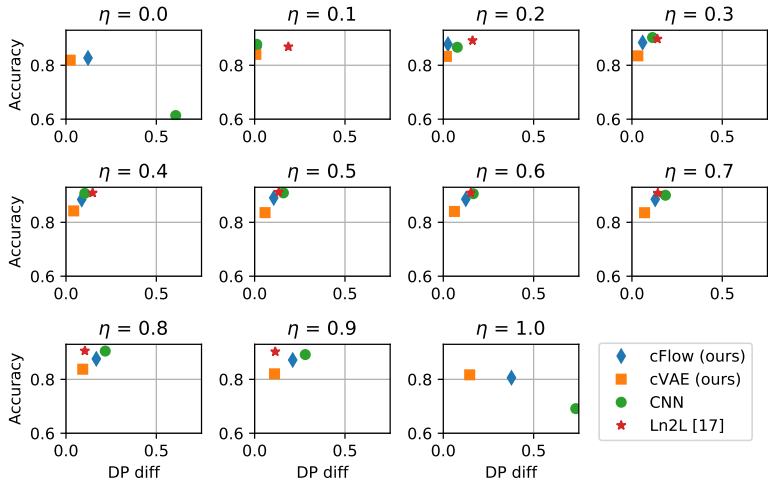


Figure 5.3: Performance of our model for the target "smiling" for different mixing factors η. *DP diff* measures fairness with respect to demographic parity. A perfectly fair model has a *DP diff* of 0, thus the closer to top-left the better it is in terms of we accuracy-fairness trade-off. Only values $\eta = 0$ and $\eta = 1$ correspond to the scenario of a strongly biased training set. The results for $0.1 \leq \eta \leq 0.9$ are to confirm that our model does not harm performance for non-biased training sets.

CelebA and UCI Adult, on the other hand, we have to generate the split from the existing data. To this end, we first set aside a randomly selected portion of the dataset from which to sample the biased dataset The portion itself is then split further into two parts: one in which $(s = -1 \wedge y = -1) \vee (s = +1 \wedge y = +1)$ holds true for all samples, call this part $\mathcal{D}_{eq}$, and the other part, call it $\mathcal{D}_{opp}$, which contains the remaining samples. To investigate the behaviour at different levels of correlation, we mix these two subsets according to a mixing factor η. For $\eta \leq \frac{1}{2}$, we combine (all of) $\mathcal{D}_{eq}$ with a fraction of 2η from $\mathcal{D}_{opp}$. For $\eta > \frac{1}{2}$, we combine (all of) $\mathcal{D}_{opp}$ and a fraction of $2(1-\eta)$ from $\mathcal{D}_{eq}$. Thus, for $\eta = 0$, the biased dataset is just $\mathcal{D}_{eq}$, for $\eta = 1$ it is just $\mathcal{D}_{opp}$ and for $\eta = \frac{1}{2}$ the biased dataset is an ordinary subset of the whole data. The test set is simply the data remaining from the initial split.

5.5.2 *Evaluation protocol*

We evaluate our results in terms of accuracy and fairness. A model that perfectly decouples its predictions from s will achieve near-uniform accuracy across all biasing-levels. For binary s/ywe quantify the fairness of a classifier's predictions using *demographic parity* (DP): the absolute difference in the probability of a positive prediction for each sensitive group.

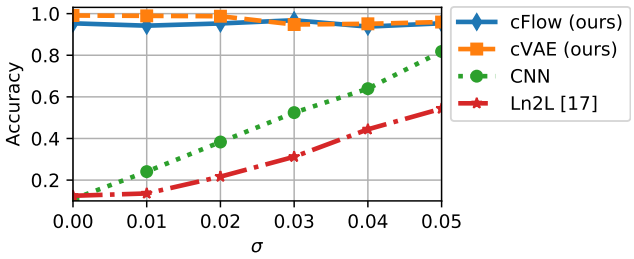


Figure 5.4: Accuracy of our approach in comparison with other baseline models on the cMNIST dataset, for different standard deviations (σ) for the colour sampling.

(a) Samples from the cMNIST training set, $\sigma = 0$.

(b) x_u null-samples from the cFlow model.

(c) x_b null-samples from the cFlow model.

Figure 5.5: Sample images from the coloured MNIST dataset problem with 10 predefined mean colours. (a): Images from the spuriously correlated subpopulation where colour is a reliable signal of the digit class-label. (b-c): Results of running our approach realised with cFlow on the cMNIST dataset. The model learns to retain the shape of the digit shape while removing the relationship with colour. A downstream classifier is now less prone to exploiting correlations between colour and the digit label class.

5.5.3 *Experimental results*

We report the results from two image datasets. cMNIST, a synthetic dataset, is a good starting point for evaluating our model due to the direct control we have over the biasing. CelebA, on the other hand, is a more practical and challenging example. We also test our method on a tabular dataset, the Adult dataset.

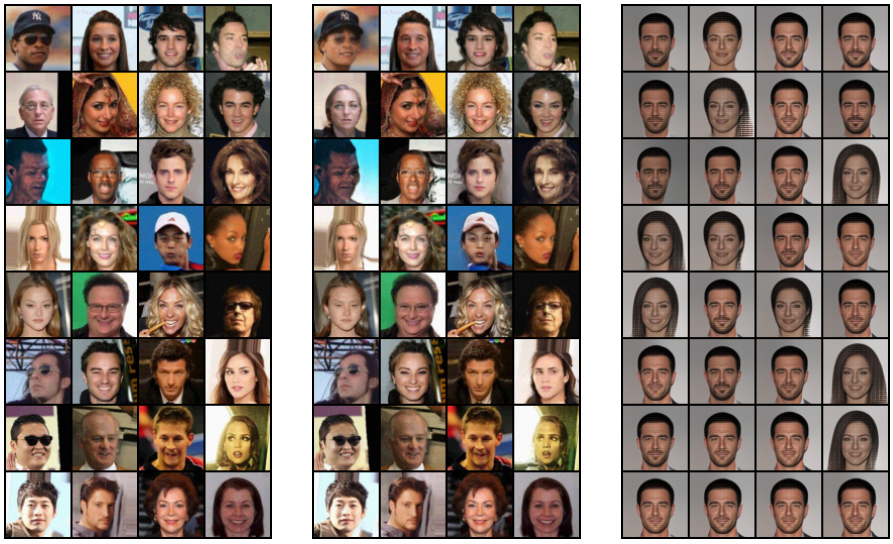
cMNIST. The coloured MNIST (cMNIST) dataset is a variant of the MNIST dataset in which the digits are coloured. In the training set, the colours have a one-to-one correspondence with the digit class. In the test set (and the representative set), colours are assigned randomly. The colours are drawn from Gaussians with 10 different means. We follow the colourisation procedure outlined by Kim et al. (2019), with the mean colour values selected so as to be maximally dispersed. The full list of such values can be found in section 5.7.5.

We produce multiple variants of the cMNIST dataset corresponding to different standard deviations σ for the colour sampling: $\sigma \in \{0.00, 0.01, ..., 0.05\}$.

For this specific dataset, we can establish an additional baseline by simply grey-scaling the dataset which only leaves the luminosity as spurious information. We also evaluate the model, with all the associated hyperparameters, from Kim et al. (2019). The only difference between the setups is the dataset creation, including the range of σ values we consider. Our versions of the dataset, on the whole, exhibit much stronger colour bias, to the point of the mapping the digit's colour and class being bijective. Fig. 5.4 shows that the model significantly underperforms even the naïve baseline, aside from at $\sigma = 0$, where they are on par.

Inspection of the null-samples shows that both the cVAE and cFlow model succeed in removing almost all colour information, which is supported quantitatively by fig. 5.4, and qualitatively by fig. 5.5. While the cVAE outperforms cFlow marginally at low σ values, performance degrades as this increases. This highlights the problems with the conditional decoder we anticipated in section 5.4.3. The lower σ, and therefore the variation in sampled colour, is, the more reliably the s label, corresponding to the mean of RGB distribution, encodes information about the colour. For higher σ values, the sampled colours can deviate far from the mean and so the encoder must incorporate information about s into its representation if it is to minimise the reconstruction loss. cFlow, on the other hand, is consistent across σ values.

CelebA. To evaluate the effectiveness of our framework on real-world image data we use the CelebA dataset (Liu et al., 2015), consisting of 202,599 celebrity images. These images are annotated with various binary physical attributes, including "gender", "hair colour", "young", etc, from which we select our sensitive and target attributes. The images are centre cropped and resized to 64×64, as is standard practice. For our experiments, we designate "gender" as the sensitive attribute, and "smiling" and "high cheekbones" as target attributes. We chose gender as the sensitive attribute as it a common sensitive attribute in the fairness literature. For the target attributes, we chose attributes that are harder to learn than gender and which do not correlate too strongly with gender in the dataset ("wearing lipstick" for example being an attribute too closely correlated with gender). The model is trained on the representative set (normal subset of CelebA) and is then used to encode the artificially biased training set and the test set. The results for the most strongly biased training set ($\eta = 0$) can be found in fig. 5.2. Our method outperforms the baselines in accuracy and fairness.



(a) Original images.

(b) $\boldsymbol{x}_u$ null-samples from the cFlow model.

(c) $\boldsymbol{x}_b$ null-samples from the cFlow model.

Figure 5.6: CelebA null-samples learned by our cFlow model, with gender as the sensitive attribute. (a) The original, untransformed samples from the CelebA dataset (b) Reconstructions using only information unrelated to s. (c) Reconstruction using only information related to $\neg s$. The model learns to disentangle gender from the non-gender related information. Note that some attributes like skin tone seem to change along with gender due to the correlation between the attributes. This is especially visible in images (1,1) and (3,2). Only because our representations are produced in the data-domain can we easily spot such instances of entanglement.

We also assess performance for different mixing factors (η) which correspond to varying degrees of bias in the training set (see fig. 5.3). This is to verify that the model does not *harm* performance when there is not much bias in the training set. For these experiments, the model is trained once on the representative set and is then used to encode different training sets. The results show that for the intermediate values of η, our model incurs a small penalty in terms of accuracy, but at the same time makes the results *fairer* (corresponding to an accuracy-fairness trade-off). Qualitative results can be found in fig. 5.6 (images from cVAE can be found in section 5.7.7).

To show that our method can handle multinomial, as well as binary, sensitive attributes, we also conduct experiments with s = hair colour as a ternary attribute ("Blonde", "Black", "Brown"), excluding "Red" because of the paucity of samples and the noisiness of their labels. The results for these experiments can be found in section 5.7.3.

RESULTS FOR THE UCI ADULT DATASET. The UCI Adult dataset consists of census data and is commonly used to evaluate models focused on algorithmic fairness. Following convention, we designate "gender" as the sensitive attrib-

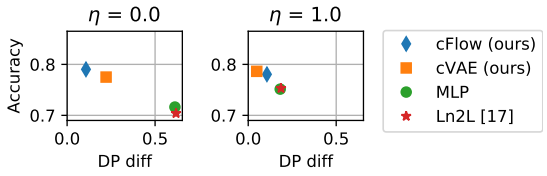


Figure 5.7: Results for the ADULT dataset. The x-axis corresponds to the difference in positive rates. An ideal result would occupy the TOP-LEFT.

ute s and whether an individual's salary is \$50,000 or greater as y. We show the performance of our approach in comparison to baseline approaches in fig. 5.7. We evaluate the performance of all models for mixing factors (η) 0 and 1. Results shown in fig. 5.7 show that we match or exceed the baseline. In terms of fairness metrics, our approach generally outperforms the baseline models for both of η. Detailed results can be found in section 5.7.3.

We also did experiments to show that the encoder transfers to other tasks. These transfer-learning experiments can be found in section 5.7.8.

5.6 Conclusion

We have proposed a general and straightforward framework for producing invariant representations, under the assumption that a representative but partially-labelled *representative* set is available. Training consists of two stages: an encoder is first trained on the representative set to produce a representation that is invariant to a designated spurious feature. This is then used as input for a downstream task-classifier, the training data for which might exhibit extreme bias with respect to that feature. We train both a VAE- and INN-based model according to this procedure, and show that the latter is particularly well-suited to this setting due to its losslessness. The design of the models allows for representations that are in the data domain and therefore exhibit meaningful invariances. We characterise this for synthetic as well as real-world datasets for which we develop a method for simulating sampling bias.

Acknowledgements

This work was in part funded by the European Research Council under the ERC grant agreement no. 851538. We are grateful to NVIDIA for donating GPUs.

Table 5.1: INN architecture used for each dataset.

Dataset	Levels	Level depth	Coupl. chan.	Input to discr.
UCI Adult	1	1	35	Null-samples
cMNIST	3	16	512	Encodings
CelebA	3	32	512	Encodings

5.7 Appendix

5.7.1 *Model Architectures*

For both cMNIST and CelebA we parameterise the coupling layers with the same convolutional architecture as in Kingma and Dhariwal (2018), consisting of 3 convolutional layers each with 512 filters of, in order, sizes 3×3, 1×1, and 3×3. Following Ardizzone et al. (2019), we Xavier initialise all but the last convolutional layer of the s and t sub-networks which itself is zero-initialised so that the coupling layers begin by performing an identity transform. We used a Glow-like architecture (Kingma and Dhariwal, 2018) (affine coupling layers together with checkerboard reshaping and invertible 1×1 convolutions) for the convolutional INNs. Table 5.1 summarises the INN architectures used for each dataset.

For the image datasets each level of the cVAE encoder consists of two gated convolutional layers (Oord et al., 2016) with ReLU activation. At each subsequent level, the number of filters is doubled, starting with an initial value 32 and 64 in the case of CelebA and cMNIST respectively. In the case of the Adult dataset, we use an encoder with one fully-connected hidden layer of width 35, followed by SeLU activation (Klambauer et al., 2017). For both cMNIST and CelebA, we downsample to a feature map with spatial dimensions 8×8, but with 3 and 16 channels respectively. For the Adult dataset, the encoding is a vector of size 35. The output layer specifies both the parameters (mean and variance) of the representation's distribution. In all cases the KL-divergence is computed with respect to a standard isotropic Gaussian prior. Details of the encoder architectures can be found in table 5.2. The loss pre-factors were sampled from a logarithmic scale; without proper balancing the networks can exhibit instability, especially during the early stages of training.

Table 5.2: cVAE encoder architecture used for each dataset. The decoder architecture in each case mirrors that of its encoder counterpart through use of transposed convolutions. For the adult dataset we apply ℓ_2 and cross-entropy losses to the reconstructions of the continuous features and discrete features, respectively.

Dataset	Initial channels	Levels	β	Recon. loss
UCI Adult	35	–	0	ℓ_2 + CE
cMNIST	32	4	0.01	ℓ_2
CelebA	32	5	1	ℓ_1

5.7.2 *Instructions for potential users*

The first question a potential user has to ask themselves is whether the method is a good fit: is the problem that the user faces one of strong spurious correlation and is there non-spurious data available that has labels for the spurious variable? To investigate the first part of the question, the user should first try to train a standard neural network classifier and observe the test-set performance. Furthermore, one should check whether the spurious variable can be removed with data augmentations alone.

If the features of the data are categorical instead of continuous, it is best to first produce a continuous representation with an autoencoder. This step only has to be done once at the beginning.

The next question is whether to use the cFlow or cVAE variant of the method. For initial experiments, we would recommend the cVAE model as it is quicker to train, and will lead to shorter feedback cycles when validating the code. If the computational budget allows it, we would recommend switching to the cFlow model once cVAE is working as it provides better guarantees regarding the retention of information from the input data.

For choosing the network architecture, the only advice we have is to look at what architectures other people have used for similar data. Note, however, that encoder-decoder architectures usually differ in some ways from classification architectures, due to their different goals: the goal of the former is primarily to compress and disentangle, while the latter aims to *discard* information unrelated to the prediction task. As such, certain layer types, like pooling layers and batch normalisation, are only suitable for classifiers and not encoder-decoders. For the INN architecture, the most import advice is to keep in mind that each individual layer is much less expressive than non-invertible layers, and so the number of layers required in INNs is much higher. However, the number also should not be *too high* or the model will overfit. It is likely that the architecture needs to be adapted during training.

See also section 5.7.4 and the code we published alongside this paper to get inspiration for architectures.

During training, the user should mostly keep an eye on two variables: the reconstruction loss and the degree of invariance of $\boldsymbol{z}_u$ w.r.t. s, which can either be gleaned from looking at reconstructions of $\boldsymbol{z}_u$ or from computing the accuracy of a downstream classifier trained on $\boldsymbol{z}_u$. The information inherent in the reconstruction loss can also be obtained by looking at full reconstructions of $\boldsymbol{z}$. If the reconstruction loss does not go down during training, some possible reasons are: the dimension of the representation is too small, the reconstruction loss weight is too small or the network just needs to be trained for longer. If the degree of invariance does not increase during training, some possible reasons are: the network is not expressive enough (e. g. not deep enough) to disentangle $\boldsymbol{z}_u$ and $\boldsymbol{z}_b$, the adversary is not powerful enough, the adversarial loss weight is too small or the network just needs to be trained for longer.

For INN training, there is the additional complication that it can become non-invertible due to numerical problems (e. g. division by zero). If this happens, the losses will quickly diverge and further training will become pointless. See section 5.7.6 for some ways of preventing this.

5.7.3 *Additional results*

DETAILED RESULTS FOR UCI ADULT DATASET. This census data is commonly used to evaluate models focused on algorithmic fairness. Following convention, we designate "gender" as the s and whether an individual's salary is \$50,000 or greater as y. We show the performance of our approach in comparison to baseline approaches in figure 5.8. We evaluate the performance of all models for mixing factors (η) of value $\{0, 0.1, ..., 1\}$. Results shown in figure 5.8 show that whilst our model fails to surpass the baseline models in terms of accuracy for the balanced case (and those close to it), we match or exceed the baseline as η moves the dataset to a more imbalanced setting. In terms of fairness metrics, our approach generally outperforms the baseline models regardless of η.

MULTINOMIAL SENSITIVE ATTRIBUTES. In addition to binary sensitive attribute s, we also investigate multinomial s in the CelebA dataset. First, we do experiments with hair colour, where s has three possible values: blond hair, brown hair and black hair. The other experiment is with a combination of age and gender, where s has four possible values, each of which is a combination of a gender and an age: Young/Female, Young/Male, Old/Female and

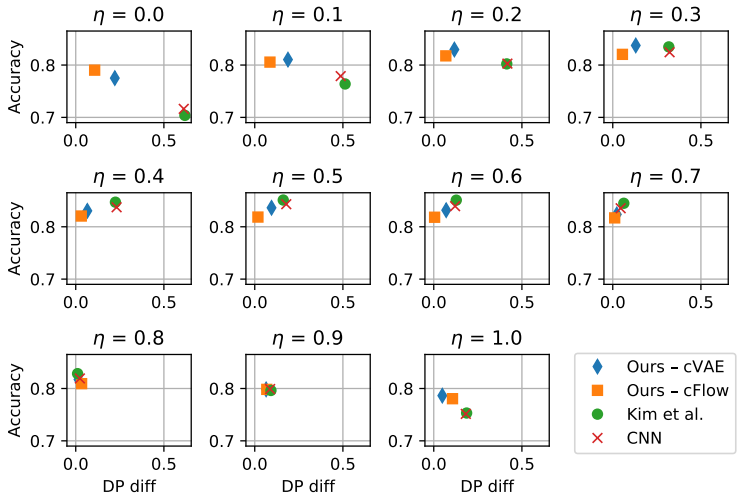


Figure 5.8: Results for the ADULT dataset. The x-axis corresponds to the difference in positive rates. An ideal result would occupy the TOP-LEFT.

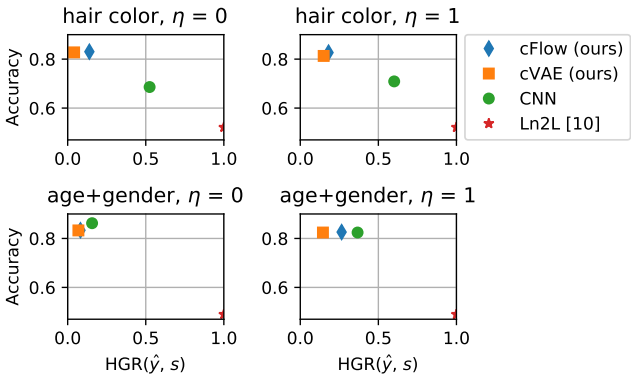


Figure 5.9: For *hair colour*, s takes on the values Blond, Brown and Black. For *age+gender*, s takes on the values Young/Female, Young/Male, Old/Female and Old/Male.

Old/Male. To evaluate the fairness for multinomial s, we use the Hirschfeld-Gebelein-Rényi Maximum Correlation Coefficient (HGR) (Mary et al., 2019) that is defined on the domain $[0, 1]$ and gives $\text{HGR}(Y, S) = 0$ iff $Y \perp S$ and 1 if there is a deterministic function to map between them. Results can be found in figure 5.9.

INVESTIGATION INTO THE SIZE OF z_b. In the cFlow model, the size of z_b is an important hyperparameter which can affect the result significantly. Here we investigate the sensitivity of the model to the choice of z_b size. Table 5.3 shows accuracy and fairness (as measured by *DP diff*) for different sizes of z_b. The results show that both too large and too small z_b is detrimental. However, they also show that the model is not overly sensitive to this parameter: both sizes 5 and 10 achieve nearly identical results.

Table 5.3: Results on the CelebA dataset with different sizes of z_b.

$\lvert z_b \rvert$	$\lvert z_b \rvert / \lvert z \rvert$	Accuracy	DP diff
1	0.0082%	0.60	0.63
3	0.0245%	0.60	0.63
5	0.0410%	0.84	0.12
10	0.0820%	0.84	0.12
30	0.2442%	0.74	0.23
50	0.4070%	0.68	0.27

Table 5.4: Additional fairness metrics for the experiments on the CelebA dataset (fig. 5.3 from the main text). *TPR diff.* refers to the difference in true positive rate. *TNR diff.* refers to the difference in true negative rate. LEFT: $\eta = 0$. RIGHT: $\eta = 1$.

Method	Accuracy	DP diff	TPR diff	TNR diff
cFlow	0.83	0.10	0.15	0.25
cVAE	0.82	0.05	0.09	0.18
CNN	0.61	0.63	0.70	0.64
Ln2L	0.52	0.00	0.00	0.00

Method	Accuracy	DP diff	TPR diff	TNR diff
cFlow	0.82	0.33	0.28	0.21
cVAE	0.81	0.16	0.10	0.05
CNN	0.67	0.75	0.66	0.76
Ln2L	0.51	0.08	0.06	0.09

ADDITIONAL FAIRNESS METRICS. In addition to *DP diff*, we report here the result from other fairness measures. These results are from the same setup as those reported in the main paper. We report the difference in TPRs between the two groups (male and female), which corresponds to a measure of Equality of Opportunity, and the difference in TNRs between the two groups.

5.7.4 *Optimisation Details*

All our models were trained using the RAdam optimiser (Liu et al., 2020) with learning rates 3×10^{-4} and 1×10^{-3} for the encoder/discriminator pair and classifier respectively. A batch size of 128 was used for all experiments.

We now detail the optimisation settings, including the choice of adversary, specific to each dataset. Details of the cVAE and cFlow architectures can be found in table 5.2 and table 5.1, respectively.

UCI ADULT. For this dataset our experiment benefited from using null-samples as inputs to the adversary of the cFlow model. Unlike for the image datasets, we found a single adversary to be sufficient. This was realised as a multi-layer perceptron (MLP) with one hidden layer, 256 units wide. The INN performs a bijection of the form $f : \mathbb{R}^n \to \mathbb{R}^n$. However, the adult dataset is composed of mostly discrete (binary/categorical) features. To achieve good performance, we found it necessary to first pre-process the inputs with a pretrained autoencoder, using its encodings as the input to the cFlow model,

as well as to the adversary. The learned representations were evaluated with a logistic regression model from scikit-learn (Pedregosa et al., 2011), using the standard settings. All baseline models were trained for 200 epochs. The Ln2L (Kim et al., 2019) and MLP baselines share the architecture of the cVAE's encoder, only with a classification layer affixed.

COLOURED MNIST. Each level of the architecture used for the downstream classifier and naïve baseline alike consists of two convolutional layers, each with kernel size 3 and followed by Batch Norm (Ioffe and Szegedy, 2015) and ReLU activation. For the Ln2L baseline, we use an a setup identical to that described in Kim et al. (2019). Each level has twice the number of filters in its convolutional layer and half the spatial input dimensions as the last. The original input is downsampled to the point of the output being reduced to a vector, to which a fully-connected classification layer is applied.

To allow for an additional level in the INN (the downsampling operations requiring the number of spatial dimensions to be even), the data was zero-padded to a size of 32×32. The cVAE and cFlow models were trained for 50 and 200 epochs respectively, using ℓ_2 reconstruction loss for the former. The downstream classifier and all baselines were trained for 40 epochs. For both of our models, an ensemble of 5 adversaries was applied to the encodings, with each member taking the form of a fully-connected ResNet, 2 blocks in depth, with SeLU activation (Klambauer et al., 2017). The adversaries were reinitialised independently with probability 0.2 at the end of each epoch. While the adversaries could equally well take null-samples as input, as done for the Adult dataset, doing so requires the performing of both forward and inverse passes each iteration, which, for the convolutional INNs of the depths we require for the image datasets, introduces a large computational overhead, while also showing to be the less stable of the two approaches in our preliminary experiments.

CELEBA. The downstream classifier and naïve baseline take the same form as described above for cMNIST, but with an additional level with 32 filters in each of its convolutions at the top of the network. For this dataset we adapt the Ln2L model by simply considering it as an augmentation the naïve baseline's objective function, with the entropy loss applied to the output of the final convolutional layer. These models were again trained for 40 epochs, which we found to be sufficient for convergence for the tasks in question. The cVAE and cFlow models were respectively trained for 100 epochs and 30 epochs, using ℓ_1 reconstruction loss for the former. Compared with cMNIST, the size of the adversarial ensemble was increased to 10, the reinitialisation

probability to 0.33, but no changes were made to the architectures of its members.

THE PITFALLS OF ADVERSARIAL TRAINING. Adversarial learning has become one of the go-to methods for enforcing invariance in fair representation learning (Ganin et al., 2016) with MMD (Louizos et al., 2016) and HSIC (Quadrianto et al., 2019), being popular non-parametric alternatives. Ganin et al. (2016) proposed adversarial learning for domain adaptation problems, with Edwards and Storkey (2016) soon after making this and learning a representation promoting demographic parity. The adversarial approach carries the benefits of being both efficient and scalable to multi-class categorical variables, which many sensitive attributes are in practice, whereas the non-parametric methods only permit pair-wise comparison.

However, when realised as a neural network, the adversary is both sensitive to the values of the inputs as well as their ordering (though exchangeable architectures, such as Zaheer et al. (2017) do exist, but which sacrifice expressiveness). Thus, it can happen that the representation learner optimises for the surrogate objective of eluding the adversary rather than the real objective of expelling s-related information. Moreover, the non-stationarity of the dynamics can lead to cyclic-equilibria, irrespective of the capacity of the adversary.

When working with a partitioned latent space, this behaviour can be averted by instead encouraging z_b to be predictive of s, acting as a kind of information "sink", as in Jacobsen et al. (2018). However, this does not have the guarantee of making z_u invariant to s - there are often many indicators for s, not all of which are needed to predict the label perfectly. Training the network to convergence before taking each gradient step with the representation learner is one way one to attempt to tame the unstable minimax dynamics (Feng et al., 2019). However, this does not prevent the emergence of the aforementioned cyclicity.

We try to mitigate the aforementioned degeneracies by maintaining a diverse set of adversaries, as has shown to be effective for GAN training (Durugkar et al., 2017), and by decorrelating the individual trajectories by intermittently re-initialising them with some small probability following each iteration.

TUNING THE PARTITION SIZES. There are several ways of ensuring that the size of z_b is sufficient to capture all s dependencies, but minimal enough that information unrelated to s is maximally preserved We adopt the straightforward search strategy of, starting from some initial guess, calibrating the

Table 5.5: Mean RGB values (in practice normalised to $[0, 1]$) parameterising the Multivariate Gaussian distributions from which each digit's colour is sampled in the biased (training) dataset. In the representative and test sets, the colour of each digit is sampled from one of the specified Gaussian distributions at random.

Digit	Colour Name	Mean RGB
0	Cyan	(0, 255, 255)
1	Blue	(0, 0, 255)
2	Magenta	(255, 0, 255)
3	Green	(0, 128, 0)
4	Lime	(0, 255, 0)
5	Maroon	(128, 0, 0)
6	Navy	(0, 0, 128)
7	Purple	(128, 0, 128)
8	Red	(255, 0, 0)
9	Yellow	(255, 255, 0)

value according to accuracy attained by a classifier trained to predict s from z_b on a held-out subset of the representative set, which is measured whenever the adversarial loss plateaus. If the accuracy is above chance level then that suggests the size of the z_b partition, $|z_b|$, needs to be increased to accommodate more information about s. If the accuracy is found to be at chance level then are two possibilities: 1) $|z_b|$ is already optimal; 2) $|z_b|$ is large enough that it fully contains both information s as well as that of a portion of y. If the former is true, then perturbations around the current value allow us to confirm this; if the latter is true then decreasing the value was indeed the correct decision.

5.7.5 *Synthesising Coloured MNIST*

We use a colourised version of MNIST as a controlled setting investigate learning from biased data in the image domain. In the biased training set, each digit is assigned a unique mean RGB value parameterising the multivariate Gaussian from which its colour is drawn. These values were chosen to be maximally dispersed across the 8-bit colour spectrum and are listed in table 5.5. By adjusting the standard deviation, σ, of the Gaussians, we adjust the degree of bias in the dataset. When $\sigma = 0$, there is a perfect and noiseless correspondence between colour and digit class which a classifier can exploit. The classifier can favour the learning of the low-level spurious feature over those higher level features constituent of the digit's class. As the standard deviation increases, the sampled RGB values are permitted to drift further from the mean, leading to overlap between the samples of the colour

(a) Original images.

(b) $\boldsymbol{x}_u$ null-samples generated by the cVAE model.

(c) $\boldsymbol{x}_b$ null-samples generated by the cVAE model.

Figure 5.10: CelebA null-samples learned by our cVAE model, with gender as the sensitive attribute. (a) The original, untransformed samples from the CelebA dataset (b) Reconstructions using only information unrelated to s. (c) Reconstruction using only information related to $\neg s$. The model learns to disentangle gender from the non-gender related information. Compared with the cFlow model, there is a severe degradation in reconstruction quality due to the model trying to simultaneously satisfy conflicting objectives.

distributions and reducing their reliability as indicators of the digit class. In the test and representative sets alike, however, the colour of each sample is sampled from one of the 10 distributions randomly, such that colour can no longer be leveraged as a shortcut to predicting the digit's class.

5.7.6 *Stabilising the Coupling layers*

Heuristically, we found that applying an additional nonlinear function to the scale coefficient of the form

$$s = \sigma(f(u)) + 0.5 \tag{5.7}$$

greatly improved the stability of the affine coupling layers. Here, σ is the logistic function, which we shift to be centred on 1 so that zero-initialising f results in the coupling layers initially performing an identity-mapping.

(a) Original images.

(b) $\boldsymbol{x}_u$ null-samples generated by the cFlow model.

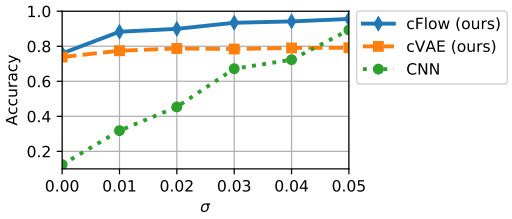
(c) $\mathbf{x}_b$ null-samples generated by the cFlow model.

Figure 5.11: CelebA null-samples learned by our cFlow model, with gender as the sensitive attribute. (a) The original, untransformed samples from the CelebA dataset (b) Reconstructions using only information unrelated to s. (c) Reconstruction using only information related to $\neg s$. The model learns to disentangle gender from the non-gender related information. Attributes such as *makeup* and *hair length* are also often modified in the process (prime examples framed with red) due to inherent correlations between them and the sensitive attribute, which the interpretability of our representations allows us to easily identify.

5.7.7 *Qualitative Results for CelebA*

Learning a representation alongside its inverse mapping, be it approximate or exact, enables us to probe the behaviour of the model that produced it, and any biases it may have implicitly captured due to entanglement between the sensitive attribute and other attributes present in the data. We highlight a few examples of such biases manifesting in the cFlow model's CelebA null-samples in fig. 5.11. In these cases, makeup and hair style have been inadvertently modified during the null-sampling due to the tight correlation between these two attributes and the sensitive attribute, gender, to which we had aimed to make our representations invariant. Additionally, in all highlighted images, the skin tone has changed: from male to gender-neutral, the skin becomes lighter and from female to gender-neutral, the skin becomes darker; in the change from male to gender-neutral, glasses are also often removed. As the model cannot know that the label is meant to only refer to gender, and not to these other (correlated) attributes, the links cannot be disentangled by the model. However, the advantage of our method is that we

can at least identify such biases due to the interpretability that comes with the representations being in the data domain.



(a) Performance on cMNIST test data after pre-training on the mixed NIST dataset.

(b) Test data input to the cFlow model.

(c) $\boldsymbol{x}_u$ null-samples generated by the cFlow model.

(d) Test data input to the cVAE model.

(e) $\boldsymbol{x}_u$ null-samples generated by the cVAE model.

Figure 5.12: Results for the transfer learning experiments in which the representative set consists of colourised samples from EMNIST, KMNIST, and FashionMNIST, while the downstream dataset remains as cMNIST. (a) Quantitative results for different σ-values. (b-c) Qualitative results for the cFlow model. (d-e) Qualitative results for the cVAE model. The qualitative results provide comparisons of the images before (left) and after (right) null-sampling. Note that for some of the cVAE samples, the clarity of the digits has clearly changed due to null-sampling, serving as an explanation for the non-increasing downstream performance.

5.7.8 *Transfer Learning*

For our method, we require a representative set which follows the same distribution as that observed during deployment. Such a representative set might not always be available. In such a scenario, we can resort to using

a set that is merely *similar* to that in the deployment setting and leverage transfer learning.

One of the advantages of using an invertible architecture over conventional, *surjective* ones that we stressed in the main text is its *losslessness*. Since the transformations are necessarily bijective, the information contained in the input can never be destroyed, only redistributed. This makes such models particularly well-suited, in our minds, for transferring learned invariances: even if the input is unfamiliar, no information should be lost when trying to transform it. This works as long as only the information about s ends up in the z_b partition. If s takes a form similar to that which we pre-trained on, and can thus be correctly partitioned in the latent space, by complement we have the information about $\neg s$ stored in the z_u partition, without presupposing similarity to the $\neg s$ observed during pre-training.

TRANSFERRING FROM MIXED-NIST TO MNIST. We test our hypothesis by comparing the performance of the cFlow and cVAE models pre-trained on a mixture of datasets belonging to the NIST family, colourised in the same way as cMNIST, while the downstream train and test sets remain the same as in the original cMNIST experiments. Specifically, we create this representative set by sampling 24,000 images (to match the cardinality of the original representative set) from EMNIST (letters only) (Cohen et al., 2017), Fashion-MNIST (Xiao et al., 2017) and KMNIST (Clanuwat et al., 2018), in equal proportion. We use the same architectures for the cVAE and cFlow models as we did in the non-transfer learning setting. In terms of hyperparameters, the only change made was to the KL-divergence's pre-factor, finding it necessary to increase it to 1 to guarantee stability.

The results for the range of σ values are shown in fig. 5.12a. Unsurprisingly, the performance of both models suffers when the representative and test sets do not completely correspond. However, the cFlow model consistently outperforms the cVAE model, with the gap increasing as the bias decreases. Although some colour information is retained in the cFlow null-samples, symptomatic of an imperfect transfer, semantic information is almost entirely retained as well. Conversely, the cVAE is very much flawed in this respect; as can be seen in the bottom row of fig. 5.12a, for some samples, semantic information is degraded to the point of the digit's identity being altered. As a result of this semantic degradation, the performance of the downstream classifier is curtailed by the noisiness of the digit's identity and is relatively unchanging across σ-values, in contrast to the monotonic improvement of that achieved on the cFlow null-samples.

6 LEARNING WITH PERFECT BAGS: ADDRESSING HIDDEN STRATIFICATION WITH ZERO LABELLED DATA

Authors: Thomas Kehrenberg[1], Myles Bartlett[1], Novi Quadrianto[1] and Viktoriia Sharmanska[1]
Affiliations:
[1] Predictive Analytics Lab (PAL), University of Sussex, Brighton, UK
Note: The appendix has been included as section 6.7.

6.1 Abstract

Machine learning models are typically trained to optimise global metrics such as average classification accuracy. Hidden stratification arises when the trained models have high average performance over classes but exhibit highly variable performance across different hidden subgroups. In this paper, we consider the setting where the hidden stratification has zero class-labelled data for some subgroups. As an illustration, we have digit images labelled as "two" or "four", each class comprises "green" and "purple" subgroups. Challengingly, in the training data, twos can be any colour, but all fours are green. Without additional knowledge, it is impossible to directly control the discrepancy of the classifier's statistics for the hidden subgroups. We develop a disentanglement algorithm that decomposes a data representation into a component that captures the subgrouping factors and a component that is invariant to them based on unlabelled (deployment) data. We cluster the unlabelled data, and equalise the cluster sizes to form "perfect bags" with respect to class and subgroup information. We cast the problem of disentangling as one of distribution matching and propose an adversarial learning approach. Unlike sample-based models, we advance a discriminator to assign scores at the level of bags of samples, with a bag being deemed authentic if was drawn from the unbiased distribution. We evaluate our approach on several classification benchmarks and show that it is indeed possible to account for zero-label hidden stratification.

6.2 Introduction

Machine learning has been deployed in safety-critical applications such as medicine (e.g. Dunnmon et al., 2019), and socially important contexts such as the allocation of healthcare, education, and credit (e.g. Hurley and Adebayo, 2017; Raghavan et al., 2020). Efficiency can be improved, costs can be reduced, and personalisation of services and products can be greatly enhanced – these are some of the drivers for the widespread development and deployment of machine learning algorithms.

Algorithms such as classifiers, however, are trained from large amount of labelled data, and are typically trained to optimise *global* metrics such as average classification accuracy. In many real-world classification tasks, each labelled class consists of multiple semantically distinct subclasses, or subgroups. For example, the "dog" class label can have finer-grained intra-class variations, such as "dog indoor" and "dog outdoor". This finer-grained subgrouping information is typically unavailable/unlabelled (e.g. Nam et al., 2020; Sohoni et al., 2020). The standard training process brings about two inter-connected challenges: a) classifiers often under-perform on important hidden subgroups (*hidden stratification*) (Oakden-Rayner et al., 2020; Sohoni et al., 2020); and b) *systematic bias* (Kallus and Zhou, 2018) affects whether or not entire collections of data points appear in the training dataset, and can make the classifier unprepared for treating those subgroups in the eventual deployment setting (*residual bias*).

We are interested in hidden stratification in which, for some of the subgroups, labelled training data is only available with a certain outcome, or labelled training samples are not available at all, due to systematic bias. This can be seen as a strong sampling bias. For instance, data on loan defaults can only be collected on those loan applicants who were approved in the past (Kallus and Zhou, 2018). Here a loan decision policy specifies whether an individual will be included in the training dataset. Individuals can be thought of as belonging to a specific subgroup such as "married" or "not married" and systematic bias produced by a historical decision policy may result in the "not married" subgroup having poor, or altogether non-existent, representation. We formalise this stratification problem as a data setting where a decision policy can lead to one or more subgroups having no labelled data.

To address the problem of subgroup bias, this paper focuses on learning subgroup-invariant representations in the presence of zero-label stratification. These representations can then be used to train a classifier that generalises to the deployment setting which does not exhibit the bias of the training set. To learn the representation, a form of supervision is needed. Our source

of supervision is motivated by the observation that we want to deploy our classifier to the eventual real-world population. A deployment set will contain data points from all subgroups. We thus consider the setting where *unlabelled* data is available for learning representations that disentangle the subgroup membership from the class membership. We note, however, that the test set could be used for this purpose in a transductive setting.

We aim to convert our unlabelled data into a collection of *perfect bags* (Kleinberg et al., 2016; Chouldechova, 2017), i. e. sample sets in which the class label y and subgroup label s are independent (i. e. $y \perp s$). Making use of the terminology from *multiple-instance learning*, our batches comprise a certain number of bags which are a collection of samples. We will then use these perfect bags as the inductive bias for learning the disentangled representations. The disentangling procedure is thus in a sense *supervised*. How can we construct these perfect bags in the absence of any labelled data? We assume that the number of subgroups is known *a priori*[1]. We then apply unsupervised k-means clustering, or a *semi*-supervised clustering based on rank statistics; the latter allows incorporating annotations from the training data when forming the clusters. Once the clusters have been found, we can sample from each cluster at an equal rate to form balanced (i. e. *perfect*) bags and use them as input for learning a disentangled representation. We cast the problem of disentangling as one of distribution matching and propose an adversarial learning approach. In the standard GAN setting, the discriminator assigns a score to each sample corresponding to the perceived probability that it was drawn from the true data distribution, and not the generator's. In contrast, we train a discriminator to assign scores at the level of bags of samples, with a bag being deemed authentic if it was drawn from the originally unbiased distribution and not the de-biased one (with the de-biaser playing the role of the generator). To do so, we take inspiration from set-classification and multiple-instance learning and equip the discriminator with a learnable attention mechanism to model interdependencies between samples in a bag.

Specifically, our paper provides the following contributions:

1. An example of systematic bias leading to one or more subgroups having *zero labelled data*.

2. Applying clustering methods to the task of transforming an *unlabelled dataset* into perfect bags.

1 Relaxing this assumption represents a clear avenue for future work. We elaborate this in the limitation and intended use sec. 6.4.3.

3. Theoretical and experimental justification that the disentangling model with *the perfect bag as an inductive bias* provides a well-disentangled representation, where one component captures the subgrouping factors and another component is invariant to them.

4. A new parametric approach to disentangling that combines elements from adversarial learning and set-classification to guide an encoder network towards the goal of producing encodings invariant to the source distribution and thereby the subgroup factors in which source distributions differ.

6.3 Related work.

We describe related work in two areas: zero-shot learning and semi-supervised learning.

6.3.1 *On zero-shot learning.*

The setting with incomplete training data, where we aim to account for seen and unseen outcomes is also known as *generalised zero-shot learning*. Traditionally, zero-shot learning transfers knowledge from classes for which we have training data to classes for which we do not, via auxiliary knowledge, e.g. via prototype examples (Larochelle et al., 2008), intermediate class description such as semantic attributes (Lampert et al., 2009; Xian et al., 2018), word2vec embeddings (Bucher et al., 2019). Our method similarly uses a collection of perfect bags as a source of auxiliary knowledge but in contrast to generalised zero-shot learning, our perfect bag is an unlabelled pool of data, where class descriptions are unknown.

6.3.2 *On semi-supervised learning.*

Wick et al. (2019) proposed a semi-supervised method that can successfully harness unlabelled data to correct for the selection bias and label bias in the training data. The unlabelled data, despite not containing the class label y, *is* labelled in terms of the subgroup label s. Our setting is significantly harder because there is no label information about y and s in the perfect bag.

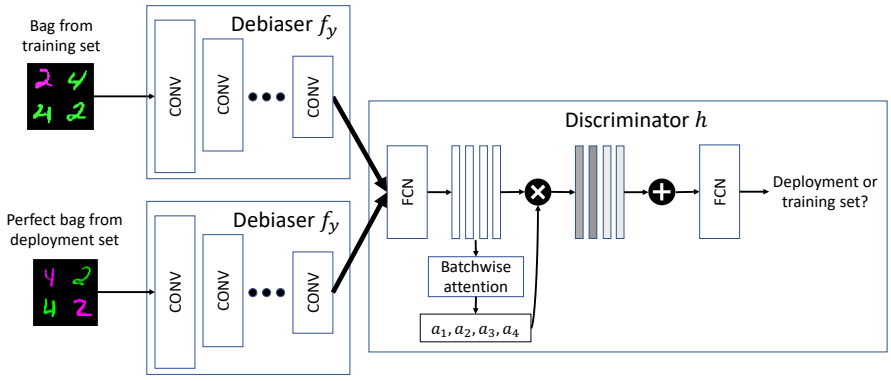


Figure 6.1: The main components involved in our proposed disentangling method, f_y (debiaser) and h (discriminator). The debiaser is trained to produce encodings, z_y of the data that are invariant to the source dataset and thereby the subgroups identifying it. In order to determine whether a bag of encodings originates from the training set or the deployment set, the discriminator performs an attention-weighted aggregation over the bag dimension to model interdependencies between the samples. In the case of Coloured MNIST where purple fours constitute the missing subgroup, the discriminator can identify an encoding of a bag from the training set by the absence of such samples so long as colour information is detectable in z_y, serving as an error signal for the debiaser.

6.3.3 *On disentangled representations learning.*

Locatello et al. (2019a) suggested that disentanglement in representation learning may be a useful property to remove algorithmic bias when subgroup information is not observed. In order for disentangled representations to reduce algorithmic bias without the knowledge of subgroup label s, they have to assume that the class label y and the subgroup label s are independent, i.e. $y \perp s$. Though, in many real-world tasks, the variable s is correlated with the variable y, and therefore unsupervised methods are not suitable (Jaiswal et al., 2018, 2019). Indeed, experiments in Locatello et al. (2019a) were wholly done with procedurally generated synthetic datasets involving 2D and 3D shapes. Without some supervision or inductive bias, disentangled representation methods would not solve the issue of algorithmic fairness with invisible demographics (Locatello et al., 2019b). Locatello et al. (2020) suggest that that it is possible to learn disentangled representations with contrasting pairs that share at least one of the underlying factors but differ in some. Here, the difference between the pairs acts as the supervision signal.

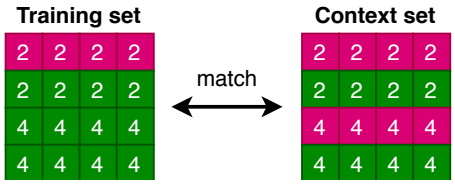


Figure 6.2: Illustration of distribution matching. In this example, the training set is lacking purple 4's. By enforcing the subspace z_y to have the same distribution for both the training and deployment set, the model is encouraged to learn a representation that is invariant to colour (or s in general.) For this to work, it is crucial that the bags be approximately balanced (perfect bags).

6.4 Methodology

6.4.1 *Theoretical background*

In this section, we first formalise the problem of hidden stratification with zero data (zero-label stratification) and the related issue of algorithmic bias. We then theoretically motivate the idea of perfect bags for reducing algorithmic bias, and their use as an inductive bias for disentanglement.

Zero-label stratification and algorithmic bias. Let S denote a set of discrete-valued subgroup labels of the associated domains $\mathcal{S}$. X, with the associated domain $\mathcal{X}$, represents other attributes of the data. Let $\mathcal{Y}$ denote the space of class labels for a classification task; $\mathcal{Y} = \{0, 1\}$ for binary classification or $\mathcal{Y} = \{1, 2, \ldots, C_{\text{cls}}\}$ for multi-class classification. For ease of exposition, we assume that we have multiple sources Ω of samples, one for each combination of class-label y and subgroup-label s. That is, we have:

$$\Omega_{y=y',s=s'}, \qquad \forall y' \in \mathcal{Y}, \forall s' \in \mathcal{S}, \tag{6.1}$$

where, for example, the source $\Omega_{y=0,s=0}$ supplies all data points with class label $y = 0$ and subgroup label $s = 0$. As in a standard supervised learning task, we have access to a labelled training set $\mathcal{D}_{tr} = \{(x_i, s_i, y_i)\}$, that is used to learn a model $M : \mathcal{X} \to \mathcal{Y}$. $\mathcal{D}_{tr}$ is composed of several sources, but lacks samples from some of the sources:

$$\exists y' \in \mathcal{Y}, \exists s' \in \mathcal{S} \; : \; \mathcal{D}_{tr} \cap \Omega_{y=y',s=s'} = \emptyset. \tag{6.2}$$

For example, we might be missing samples from two sources: $\Omega_{y=0,s=0}$ and $\Omega_{y=1,s=0}$. In binary classification, this corresponds to no labelled data for the subgroup label $s = 0$, a setting we refer to as *missing subgroup* (MS). Other times, we may observe a one-sided (negative) outcome for the subgroup label

$s = 0$ (i. e., we have $\mathscr{D}_{tr} \cap \Omega_{y=1,s=0} = \emptyset$), giving rise to a setting we refer to as *subgroup bias* (SB).

Once the model M is trained, we deploy it to the diverse real-world data. That is, it will encounter data which has overlap with all sources. If the model relies only on the incomplete training set, it is to be expected that the model will misclassify the subgroups with zero training data. The model becomes biased against those subgroups, leading to unexpectedly poor performance when it is deployed.

We propose to alleviate the issue of bias against missing subgroups by mixing labelled data with unlabelled data that is usually much cheaper to obtain (Chapelle et al., 2006). In this paper, we refer to this set of *unlabelled* data as the deployment set[2] $\mathscr{D}_{dep} = \{(x_i)\}$. This deployment set has overlap with all sources:

$$\mathscr{D}_{dep} \cap \Omega_{y=y',s=s'} \neq \emptyset \qquad \forall y' \in \mathscr{Y}, \forall s' \in \mathscr{S}\,. \tag{6.3}$$

Importantly, the deployment set has no information about class labels y or the subgroup labels s.

RELATION TO ALGORITHMIC FAIRNESS. Training set bias also affects what has been termed *algorithmic fairness*, which is commonly expressed in terms of the predicted class $\hat{y}$ of a machine learning model M. We adopt a statistical notion of algorithmic fairness in which outcomes are balanced under certain conditions between groups of data points with different subgroup labels. Several statistical bias measures have been proposed (Kamiran and Calders, 2012; Hardt et al., 2016; Chouldechova, 2017; Zafar et al., 2017a; Raghavan et al., 2020) (shown below for the case where s and y are binary):

$$P(\hat{y} = 1|s = 0) = P(\hat{y} = 1|s = 1) \tag{6.4}$$

$$P(\hat{y} = 1|s = 0, y) = P(\hat{y} = 1|s = 1, y) \tag{6.5}$$

$$P(y = 1|s = 0, \hat{y}) = P(y = 1|s = 1, \hat{y}) \tag{6.6}$$

(6.4) is equality of positive rate; (6.5) is equality of true positive/negative rate; (6.6) is equality of positive/negative predicted value. Generally, these statistical notions can be expressed in terms of different (conditional) independence statements between the involved random variables (Barocas et al., 2019): $\hat{y} \perp s$ (equation 6.4), $\hat{y} \perp s \mid y$ (equation 6.5), and $y \perp s \mid \hat{y}$ (equation 6.6). If our training set has no positive outcome for the subgroup label $s = 0$, i.e. $\Omega_{y=1,s=0} = \emptyset$, the true positive rate for this subgroup will suffer, and

2 In our experiments, we report accuracy and bias metrics on another independent test set instead of on the unlabelled data that is available at training time.

therefore we will likely not be able to satisfy, among others, equality of true positive rate. In the experimental section, we use metrics based on these equalities to quantify how strongly the predictions are affected by the dataset bias.

PERFECT BAG. We call a sampled set for which $y \perp s$ holds, a perfect bag (Kleinberg et al., 2016; Chouldechova, 2017). Such sets are also very desirable when training algorithmically fair classifiers (see the *sampling* method in Kamiran and Calders, 2012). Ideally, we would like to sample our deployment dataset as perfect bags. However, the deployment set is unlabelled and is unlikely to be perfect in practice. Instead, we pursue learning under zero-label systematic bias as learning disentangled representations with a collection of *approximately* perfect bags (produced with clustering techniques, see section 6.4.2). We show that the disentangling procedure is robust enough to work with this relaxation, but that performance scales with how well the deployment set is balanced.

DISENTANGLED REPRESENTATION. Disentanglement-learning aims to find a factorised representation of a data point x through mapping functions f_i such that $f_i(x) = z_i$ where $z_1, z_2, \dots, z_p$ are p distinct (independent) factors of variations, which together form x. We can formalise this intuitive definition using group and representation theories (Higgins et al., 2018), or using structural causal models (Suter et al., 2019). Specifically for this paper, we would like to split the data representation into two factors as $f_y(x) = z_y$ and $f_s(x) = z_s$ where z_y contains factors that are relevant for y-prediction and z_s contains factors related to the subgroup label s. Since s is correlated with the class label y, we need annotations of the undesired nuisance variable s (Jaiswal et al., 2018, 2019) to be successful in using disentanglement learning methods for zero-label stratification. We have some annotations of subgroup label s in the training set $\mathscr{D}_{tr} = \{(x_i, s_i, y_i)\}$, however, crucially, due to systematic bias, this set is missing certain subgroups. We have all subgroups in the deployment set $\mathscr{D}_{dep} = \{(x_i)\}$, though, the challenge is that the subgrouping information is unavailable or hidden at the deployment time. In the following section, we show that we can still leverage the deployment set for learning the disentangled representations.

DISENTANGLEMENT WITH A COLLECTION OF PERFECT BAGS. Our framework for learning the disentangled representations comprises four core modules: 1) *encoder* functions f_y and f_s (which share weights) that embed x into z_y and z_s, respectively; 2) a *decoder* function g that learns the approximate-inverse

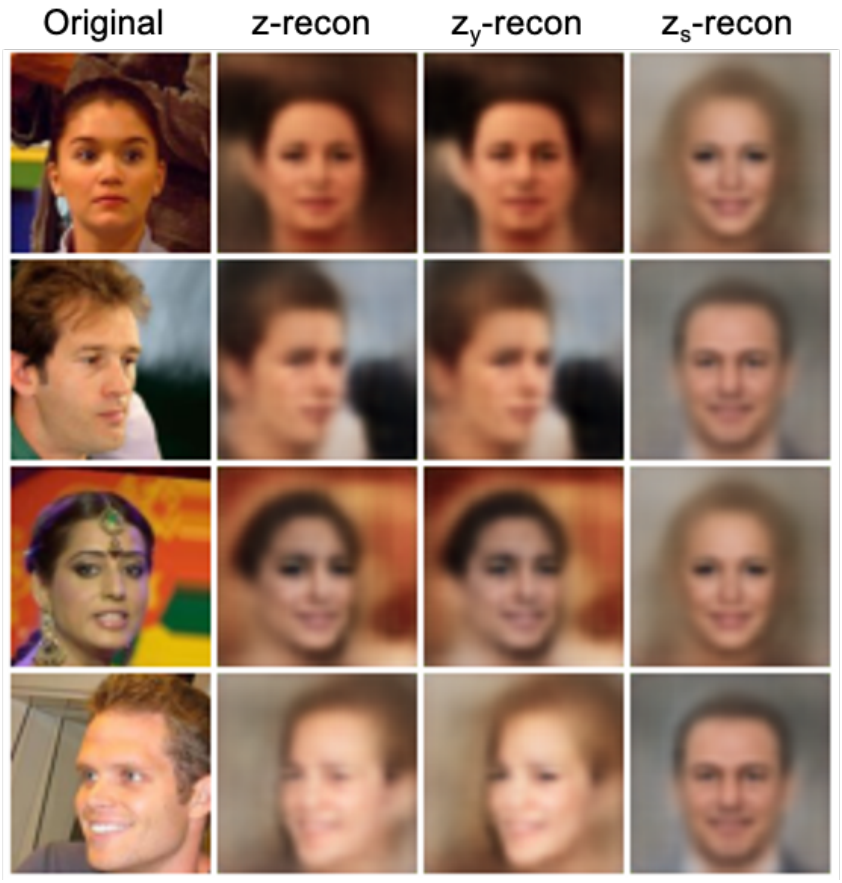


Figure 6.3: Visualisation of our method's solutions for the CelebA dataset, with"smiling females" as the missing subgroup. Column 1 shows the original images from x from the deployment set of CelebA. Column 2 shows plain reconstructions generated from $x_{recon} = g(f_y(x), f_s(x))$. Column 3 shows reconstruction with zeroed-out z_s: $g(f_y(x), 0)$, which effectively visualises z_y. Column 4 shows the result of an analogous process where z_y was zeroed out instead.

mapping of f_y and f_s: $g : (z_y, z_s) \rightarrow \tilde{x}$; 3) *predictor* functions ℓ_y and ℓ_s that predict y and s from z_y and z_s respectively, and 4) a *discriminator* function h that classifies a given bag of samples embedded in z_y as deriving from the deployment set or the training set; this marks a significant departure from the typical GAN discriminator, which takes as input batches of data and yields a prediction for each sample independently of the other samples in the batch. Fig. 6.1 shows our framework. Formally, given bags $\mathscr{B}_{tr}$ from the training set, and *balanced* (i.e. approximately perfect – see section 6.4.2 for details on how this can be practically achieved) bags $\mathscr{B}_{perf}$ from the deployment

set, we first define, for notational convenience, the loss with respect to the encoder networks, f_y and f_s as

$$\begin{aligned}
&\mathscr{L}_{\text{enc}}(f_y, f_s, h) \\
&= \sum_{x \in \mathscr{B}_{tr} \bigcup \mathscr{B}_{perf}} L_{\text{recon}}(x, g(f_y(x), f_s(x))) \\
&\quad + \sum_{x \in \mathscr{B}_{tr}} \lambda_1 L_{\text{sup}}(y, \ell_y(f_y(x))) + \lambda_2 L_{\text{sup}}(s, \ell_s(f_s(x))) \\
&\quad - \lambda_3 (\log h(\{f_y(x) | x \in \mathscr{B}_{perf}\}) \\
&\qquad + \log h(\{f_y(x) | x \in \mathscr{B}_{tr}\})),
\end{aligned} \tag{6.7}$$

where L_{recon} and L_{sup} denote the reconstruction loss, and supervised loss, respectively, and λ_1, λ_2 and λ_3 are positive pre-factors. The overall objective, encompassing f_y, f_s, and h can then be formulated in terms of $\mathscr{L}_{\text{enc}}$ as

$$\min_{f_y, f_s} \max_{h} \mathscr{L}_{\text{total}} = \mathscr{L}_{enc}(f_y, f_s, h) \,. \tag{6.8}$$

Aside from being computed over a bag of samples, our adversarial loss differs from that the standard one in that both of its constituent terms are dependent on f_y (the encoder is responsible for producing the "real" *and* "fake" samples); we allow the gradient to flow through both of these terms, finding that adding a stop-gradient to $\log h(\{f_y(x) | x \in \mathscr{B}_{perf}\})$ drastically reduced the convergence-rate and stability of the algorithm.

Eq. (6.8) is computed over batches of bags and the discriminator is trained to map a bag of data points from the training set and the deployment set to a binary label: 1 if the bag is judged to have been sampled from the deployment set, 0 if from the training set. Since the task is a set-classification one, we require that the function it defines respects the exchangeability of the bag dimension – that is, the discriminator's predictions should take into account dependencies between samples in a bag but should be invariant to the order in which they appear, i. e. we have $h(\{f_y(x_i)\}_{b=i}^{\mathscr{B}}) = h(\{f_y(x_i)\}_{b=\pi(i)}^{\mathscr{B}})$ for all permutations π. To make the entirety of the function h – composed of sub-functions $h_1(h_2(h_3...))$ – have this property, requires only the innermost, sub-function, ρ in the chain to have it. While there are a number of choices when it comes to defining ρ, we choose a weighted average $\rho = \frac{1}{\mathscr{B}} \sum_i (\{\text{attention}(f_y(x_i))\}_{b=i}^{\mathscr{B}})$, with weights computed according to a learned attention mechanism. The idea of using an attention mechanism for set-wise classification has been previously successfully explored by Ilse et al. (2018) and Lee et al. (2019); we use the gated attention mechanism proposed by Ilse et al. (2018) in the experiments, but also tried out the scaled dot-product attention per Vaswani et al. (2017), as the bag-wise pooling layer of our discriminator. The result

of ρ is then processed by a series of fully-connected layers, following the DeepSets (Zaheer et al., 2017) paradigm, which ultimately computes a single prediction for a given bag of samples.

Our goal is that z_y is invariant to the subgroup s. However, what the adversarial loss actually enforces is that z_y has the same distribution for the bags from the deployment set and the bags from the training set. To ensure that the network learns the correct task, it is crucial that subgroup membership is the only differing factor between the two types of bags. The first step towards this goal is that bags from the deployment set are balanced (or *perfect*): all combinations of s and y appear at the same rate. The second step is that, in the bags from the training set, the possible values of y have to appear at the same rate, because y is meant to be preserved; thus, $P(y_{tr} = 0) = P(y_{tr} = 1) = \ldots$. Finally, within the classes, subgroups should appear at equal rate: $P(s_{tr} = 0|y_{tr} = y') = P(s_{tr} = 1|y_{tr} = y') = \ldots$. For example, if the bag size is 4, Y and S are binary, and the combination $(y = 1, s = 0)$ is missing, then each bag should contain 2 samples of $(y = 1, s = 1)$ and 1 sample each of $(y = 0, s = 0)$ and $(y = 0, s = 1)$ (see also fig. 6.2). This ensures that the bags from the training set only differ in those samples from the deployment bags, where a subgroup is missing. To guide the network towards the desired solution, we supplement this implicit constraint with the explicit constraint that z_y be predictive of y, which we achieve using a linear predictor l. Whenever we have $\dim(\mathcal{S}) > 1$ we can also impose the same constraint on z_s, but with respect to s. With these conditions met, to fool the discriminator, the encoder must separate out information pertaining to S into the space z_s not part of the discriminator's input, leaving only subgroup-unrelated information in z_y.

In the *missing subgroup* scenario, the disentangling supervision is weaker. Thus, for this case, we restrict the amount of information that can be encoded in z_s by setting its dimensionality equal to $\lceil \log_2(\dim(\mathcal{S})) \rceil$; e. g., in the case of binary s, we restrict z_s to be one-dimensional. Additionally, z_s can be discretised with a straight-through estimator of the gradient (Bengio et al., 2013).

Note that as long as the model sees the different s-y-combinations at the right proportions, disentangling can (to an extent) also be achieved without the bag-wise loss (i. e., when the bag size is set to 1). The model then has to learn an implicit prior for which s-y-combinations to expect in the two different sets – training set and deployment set. However, in our preliminary experiments we found this sample-wise approach to work much more poorly than using a bag-wise loss. For those experiments, the batch was balanced in the same way as described above for the bags, but the batch was not

subdivided into bags and there was no attention or aggregation method applied to the batch: the loss was just computed per sample. From the poor performance, we inferred that the sample-wise loss lacks the necessary context to easily differentiate between the two datasets.

Our contribution to the disentanglement problem is thus two-fold: i) the use of the difference between two distinct sets as the supervision signal, and ii) the use of a bag-wise loss (where the bags are sampled to represent their corresponding data distribution in an idealised way) which allows the model to contrast the two sets more easily. Together, they form the idea of disentangling with "perfect bags".

As mentioned in section 6.3.3, Locatello et al. (2020) learned to disentangle from non-i.i.d. pairs which share at least one underlying factor. In our scenario, constructing the pairs each out of one sample from the training set and one sample from the deployment set would satisfy this requirement if we allow the 'underlying factor' to be quite abstract (see also the section below for a discussion of this). However, constructing the pairs each out of two *bags* instead of two samples makes the analysis much clearer: bags from the training and deployment set are constructed such that they both have the same distribution of y values (a uniform distribution over y), but they differ in the distribution of s values. Thus, the *shared* factor is the one concerning the prediction target y, and the *differing* factor is the one concerning s.

Disentanglement guarantees. Under Shu et al. (2020)'s framework, our disentanglement supervision corresponds to *match pairing*: We observe samples from a pair of distributions (deployment set and training set). As an example, we consider coloured MNIST with two classes (digits 2 and 4) and two subgroups (colours purple and green), where the training set is lacking the combination of digit 4 and colour purple. Sample pairs from the two distributions have a shared underlying factor: 2's can be purple and green. The pairs also have a factor that differs: in the training set, a 4 can only be green; in the deployment set, it can have both colours. Theorem 1 in (Shu et al., 2020) guarantees then that a (sufficiently powerful) disentangling encoder will produce a disentangling that is *consistent* with respect to the shared factor and *restricted* with respect to the changing factor. In our case, z_y will capture the shared factor and z_s the changing factor. For the invariant representation, we drop z_s, which can be seen as setting this value to 0; as z_s has guaranteed *restrictiveness*, the effect of changing z_s is restricted to changing the corresponding factor, which in the example refers to whether or not 4's can be purple.

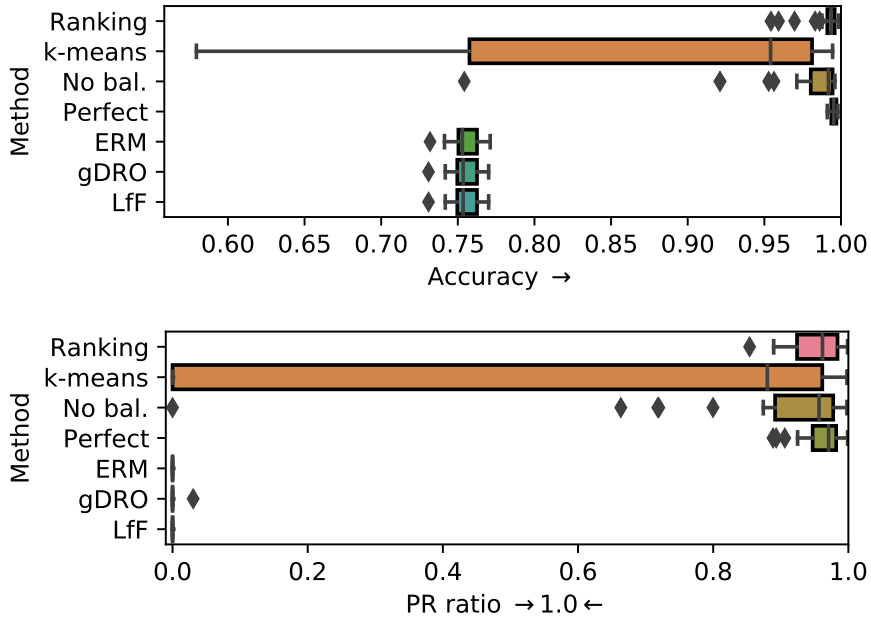


Figure 6.4: Results from 30 repeats for the Coloured MNIST dataset with two digits, 2 and 4, with *subgroup bias* for the colour 'purple': for purple, only the digit class '2' is present. LEFT: Accuracy. RIGHT: Positive rate ratio. For the `Ranking` clustering, the clustering accuracy was 96% ± 6%; for `K-means` it was 64% ± 10%.

6.4.2 *Implementation*

Our proposed framework, demonstrated in fig. 6.1, entails two steps: 1) sample perfect bags from an unlabelled deployment set, and 2) produce disentangled representations using perfect bags for adversarial distribution-matching.

CONSTRUCTING APPROXIMATELY PERFECT BAGS VIA CLUSTERING. We cluster the data points from the deployment set into $K = \dim(\mathcal{Y}) \cdot \dim(\mathcal{S})$ number of clusters, i.e. the number of data sources $\Omega_{y,s}$. We use the k-means clustering algorithm, and a recently proposed method based on rank statistics (Han et al., 2020). The cluster assignments can then be used to evenly stratify the deployment set into perfect batches, to be used by the subsequent disentangling phase.

As a result of clustering, the data points in the deployment set $\mathcal{D}_{dep}$ are labelled with cluster assignments $\mathcal{D}_{dep} = \{(x_i, c_i)\}$, $c_i = C(z_i)$. We balance $\mathcal{D}_{dep}$ so that all clusters have equal size to form a perfect bag , and use it as a supervision signal for the disentangling step.

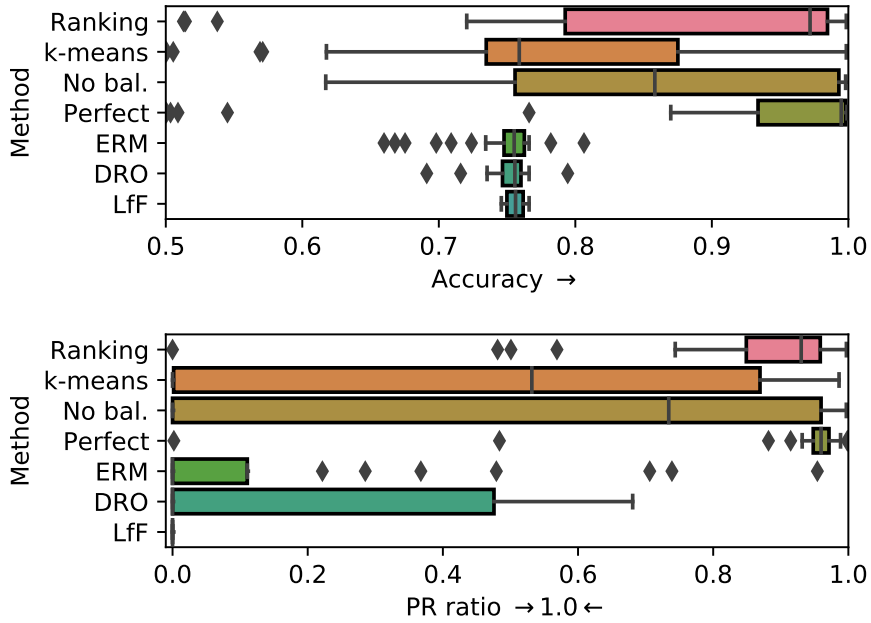


Figure 6.5: Results from 30 repeats for the Coloured MNIST dataset with two digits, 2 and 4, with a *missing subgroup*: the training dataset only has green digits. LEFT: Accuracy. RIGHT: Positive rate ratio. For the `Ranking` clustering, the clustering accuracy was 88% ± 5%; for `K-means` it was 72% ± 16%.

CLUSTERING REQUIREMENTS. For constructing the perfect bags, the correspondence between the clusters and the class-labels/subgroup-label pairs does not need to be known. The clustering is only needed for drawing an equal number of samples from each cluster for each bag of samples. In our experiments, we provide an analysis with unsupervised k-means clustering where we do not use annotations from the training set as side information, even for the *known* groups. When clustering with the training labels (such as with the rank statistics approach), we use the information that they provide to ensure samples from the *known* subgroups are clustered together with others with the same label.

CLUSTERING GUARANTEES. By leveraging the feature space of deep models, previous research has shown that semantically meaningful clusters can be discovered without the need for ground-truth annotations (for example Gansbeke et al., 2020; Han et al., 2020; Oakden-Rayner et al., 2020; Sohoni et al., 2020). Sohoni et al. (2020) recently provided a "clustering guarantee" when assuming access to ground truth class labels y and using clustering within each class to generate approximate subgroups s. Even with infinite data, accurate identification of the subgroups is unfortunately impossible. Instead, Sohoni et al. (2020) showed that the error in their quantity of interest, classification loss for each subgroup, can be bounded by the total variation

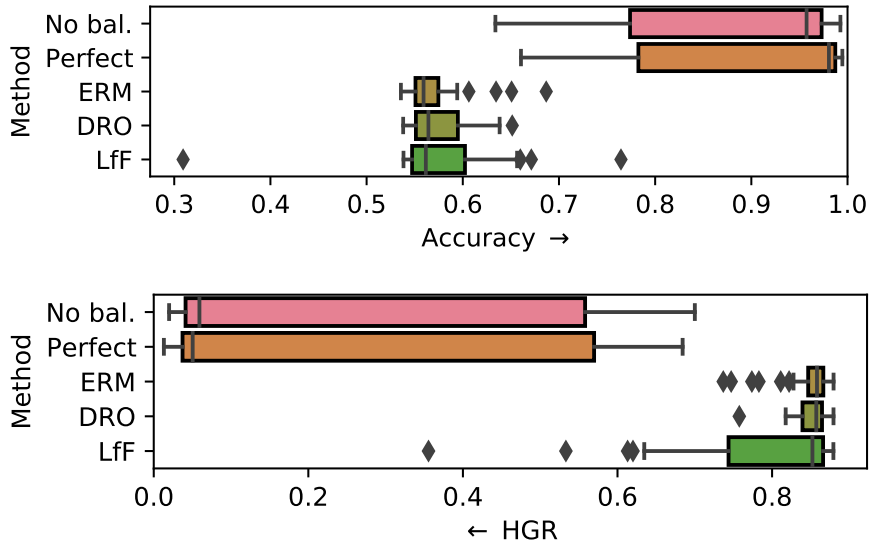


Figure 6.6: Results from 30 repeats for the Coloured MNIST dataset with three digits: '2', '4' and '6'. Four combinations of digit and colour are missing: green 2's, blue 2's, blue 4's and green 6's. LEFT: Accuracy. RIGHT: Hirschfeld-Gebelein-Rényi maximal correlation (Rényi, 1959) between S and Y.

estimation error in the mixture-of-Gaussians case. Our primary goal is to ensure good construction of perfect bags. We can adapt the analysis of Sohoni et al. (2020) to bound the error of our quantity of interest, *aggregate statistics* of the approximated perfect bag for each subgroup. We can subsequently apply the union bound to take into account that we have a fully-unlabelled setting.

6.4.3 *Limitation and intended use*

Although having zero labelled examples for some subgroups is not uncommon due to the effects of systematic bias, we should make a value-judgement on the efficacy of the dataset with respect to a task. We can then decide whether or not to take corrective action as described in this paper.

A limitation of the presented approach is that, for constructing the perfect bags used to train the disentangling algorithm, we have relied on knowing the number of clusters *a priori*, something that, in practice, is perhaps not the case. Removing this dependency through automatic determination of the number of clusters would generalise our method further but this line of research is challenging and extends beyond the scope of the current paper. One difficulty is that we need to ensure that the small but salient clusters are correctly identified. The cluster formed by an underrepresented subgroup

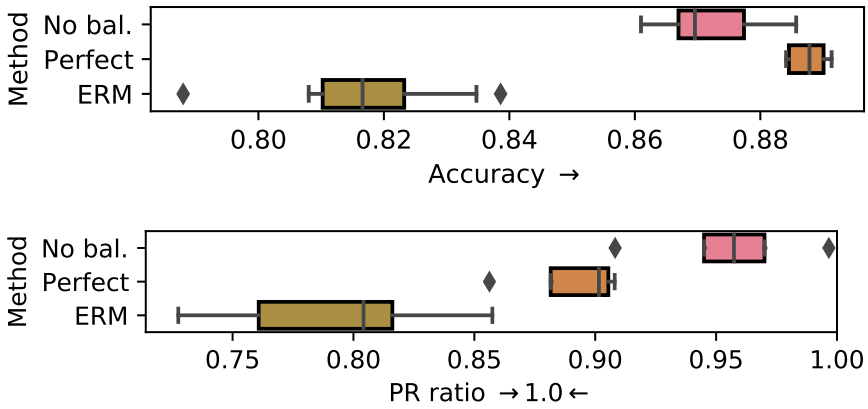


Figure 6.7: Results from 10 repeats for the CelebA dataset with the *subgroup bias* setting. The task is to predict "smiling" vs "non-smiling" and the subgroups are based on gender. The subgroup "female" is missing samples for the "smiling" class. LEFT: Accuracy. RIGHT: Positive rate ratio.

can be easily overlooked by a clustering algorithm in favour of larger but less salient clusters (which may be sub-clusters of other, larger, subgroups).

6.5 EXPERIMENTS

We perform experiments using image and tabular datasets: Coloured MNIST (Kim et al., 2019), CelebA (Liu et al., 2015) and Adult Income[3] (Dheeru and Karra Taniskidou, 2017) that are publicly available. To validate the first step of creating the perfect bags, we compare the performance of our disentangling model when paired with each of four different balancing methods: 1) with clustering via rank statistics (`Ranking`); 2) with clustering via k-means (`K-means`); 3) without balancing, when the deployment set $\mathscr{D}_{dep}$ is used as is (`No bal.`); 4) with balancing done using the ground-truth class and subgroup labels (`Perfect`) that would in practice be unobservable; this provides insight into what can be achieved under ideal conditions and how sensitive the method is to imperfections in the bag-balancing.

To validate the disentangling step, we compare with three other baselines. In all cases, the training set is balanced in the same fashion required for our method. This is similar in effect to the sampling method proposed by Kamiran and Calders (2012) but with those samples with the same class label as the missing sources upsampled, based on knowledge of $\dim(\mathscr{S})$. We denote a classifier trained with cross-entropy loss on this data as `ERM`. The second of the aforementioned baselines is `DRO` (Hashimoto et al., 2018), which functions without subgroup labels by minimising the worst-case training loss over all

3 Results for Adult Income dataset and discussion can be found in the supplementary material.

possible groups that are above a certain minimum size; along with a variant of it, `gDRO` (Sagawa et al., 2019), that exploits subgroup label information but is then only applicable when $\dim(\mathcal{S}_{tr}) > 1$. Third, we have the `LfF` model proposed by Nam et al. (2020) that works by reweighting the cross-entropy loss of a classifier using the predictions of a purposely biased sister network.

6.5.1 *Coloured MNIST*

The MNIST dataset (LeCun et al., 1998) consists of 70,000 (60,000 designated for training, 10,000 for testing) images of grey-scale hand-written digits. We colour the digits following a similar procedure to that outlined by (Kim et al., 2019), randomly assigning each sample one of ten distinct RGB colours. Each source is then a combination of digit-class (class label) and colour (subgroup label). We use no data-augmentation aside from symmetrically zero-padding the images to be of size 32x32. We create imbalance in both D_{dep} and D_{tr} by sub-sampling the remaining sources; we do this to render a more realistic setting in which the deployment set is not innately balanced and demands preliminary clustering to construct approximately perfect bags. The sub-sampling proportions used for each set of experiments can be found in the appendix.

We begin by considering a binary, 2-digit, 2-colour, variant of the dataset with $Y = \{2, 4\}$ and $S = \{\text{green}, \text{purple}\}$. For this variant we explore both the SB (subgroup bias) and MS (missing subgroup) settings. To simulate the SB setting, we set $\Omega_{y=4,s=\text{purple}}$ to be the missing source. To simulate the MS setting, where we have training data for only a single subgroup, we set both $\Omega_{y=\text{'two'},s=purple}$ and $\Omega_{y=\text{'four'},s=purple}$ to be the missing sources (i.e. D_{tr} consists of only green digits).

Fig. 6.4 shows the results for the SB setting. We see that the performance of our method directly correlates with how balanced the bags are, with the ranking of the different clustering methods being `Perfect` > `Ranking` > `No bal.` > `K-means`. Even without balancing (`No bal.`) our method greatly outperforms the baselines methods, which all exhibit similar performance to one another in terms of both accuracy and PR ratio (positive rate ratio). The PR ratio is given by $P(\hat{y}=1|s=1)/P(\hat{y}=1|s=0)$; it quantifies how invariant the classifier output $\hat{y}$ is to the subgroups. The optimal value is 1.

Fig. 6.5 shows that the problem of *missing subgroups* is harder to solve. Again we see that poor clustering (in the case of `K-means`) is detrimental to the disentangling procedure, though for all balancing strategies the IQR is significantly higher than observed in the MS setting, along with there being a number of extreme outliers. The median, however, remains high,

which is reflective of the "hit-or-miss" performance of the method under these conditions, but where the number of hits far outweighs the number of misses. Visualisation of the reconstructions with z_y zeroed-out suggest that misses often mostly occurred due to the semantic information being concentrated in z_s while z_y is left to contain only residual information, even when z_s was set to be one-dimensional and binarised. We leave it to future work to explore how to better offset such degeneracies.

To investigate how an increase in the number of classes affects disentangling of classes and groups, we look to a 3-digit, 3-colour variant of the dataset in the SB setting where four sources are missing from D_{tr}. Results for this configuration are shown in fig. 6.6. We see that the performance of `No bal.` is quite close to that of `Perfect`, we suspect this is because balancing is less critical with the increased number of subgroups strengthening the training signal. As the PR ratio is not a suitable metric for non-binary S, we instead quantify the invariance of the predictions to the subgroup with the HGR maximal correlation (Rényi, 1959).

6.5.2 *CelebA*

To demonstrate our method can be used to mitigate biased decision making for real-world computer vision problems, we consider the CelebA dataset (Liu et al., 2015) comprising over 200,000 images of different celebrities. The dataset comes with per-image annotations of attributes related to visual appearance, emotion, gender, age. We predict the smiling attribute as the class label and use the binary attribute, gender, as the subgroup label. Here, we consider an SB setting, where "smiling females", $\Omega_{s=0,y=1}$, constitutes a missing source. Since the dataset exhibits natural imbalance in $S \times Y$, we perform no additional sub-sampling of either the training set or the deployment set, as we did for Coloured MNIST. Fig. 6.7 shows the model trained with a perfectly balanced deployment set `Perfect` outperforms `No bal.`, indicating that this natural imbalance is sufficiently strong to somewhat disrupt the disentangling procedure and is something to be potentially remedied through clustering. We did not use the clustering approach for CelebA, as the attributes "smiling" and "gender" are not the most salient; more work is needed on the clustering side to discover all semantically meaningful clusters. Nonetheless, our method, with or without the artificial balancing, yields much better accuracy than that of `ERM`. Furthermore, we show qualitative results of the disentangling in fig. 6.3, and note a clear separation of subgroup-relevant information from subgroup-irrelevant information.

6.6 Conclusion

We have highlighted the problem that systematic bias can result in one or more subgroups having zero labelled data, and by doing so hope to have stimulated serious consideration for it (even if to be dismissed) when planning, building, evaluating and regulating machine learning systems. We propose a two-step approach for addressing the resulting zero-label stratification problem. First, we construct perfect bags from an unlabelled deployment set via clustering. Second, we learn a disentangled representations using the perfect bags for adversarial distribution-matching. We empirically validate our framework on the Coloured MNIST, CelebA and Adult Income datasets, and find evidence that it is possible to maintain high performance on the subgroups with zero training data. We analyse our approach using the disentanglement calculus of Shu et al. (2020) that relies on the notions of restrictiveness and consistency, and show that we can derive some guarantees. We presented our approach in the context of biased data, but it could be used as a general-purpose method for learning disentanglements as long as the factors can be expressed by contrasting two data subsets; that is, the data will be disentangled into two factors, the first of which corresponds to the aspect that is common in the two sets, and the second corresponding to the one that differs. The method thus does not compete directly with *unsupervised* disentanglement, but is potentially more widely applicable than to only fairness problems.

However, it is not really accurate to say that we found a new "fairness-based" way to learn disentanglements. Rather, it is the opposite way: we found a new disentanglement-based fairness method. In order to achieve the goal of fairness, we used building blocks (like adversarial training) which happen to also be used in disentanglement learning. And thus, it is not surprising that the method does in fact work by disentangling, but it was just the choice of tools which would get the job done that produced this outcome. The fairness aspect only acted as the motivator, and did not in itself contribute to the solution.

Another take-away from our work concerns the general disentangling setup described in Locatello et al. (2020) – i. e. using pairs that share at least one underlying factor but also differ in some. For our work, we found it advantageous to do this pair matching not on the level of samples, but on the level of bags (i. e., small, specially-sampled *sets* of samples). It is possible that other disentangling problems would benefit from a similar switch in perspective that identifies not the samples but some other structure as the natural way to construct the pairs for the disentangling.

Future work includes further restricting the amount of information that can be encoded in the bias factor, and the limitations that we mentioned in sec. 6.4.3. Furthermore, the literature on clustering can potentially provide further insights on theoretical guarantees for clustering. This was not a focus in this paper as the disentangling aspect is more crucial to the approach.

6.7 Appendix

6.7.1 *Results for Adult Income*

Figures 6.8 and 6.9 show results from our method on the Adult Income dataset Dheeru and Karra Taniskidou, 2017. This dataset is a common dataset for evaluating fair machine learning models. Each instance in the dataset is described by 14 characteristics including gender, education, marital status, number of work hours per week among others, along with a label denoting income level (≥$50K or not). We transform the representation into 62 real and binary features along with the subgroup label s. The dataset is naturally imbalanced with respect to gender: 30% of the males are labelled as earning more than $50K per year (high income), while only 11% of females are labelled as such. For further details on the dataset construction, see section 6.7.2. Following standard practice in algorithmic fairness, e.g. Zemel et al. (2013), we consider gender to be the subgroup label s.

We study the following two settings. 1) *subgroup bias*: we have labelled training data for males ($s = 1$) with both positive and negative outcomes, but for the group of females ($s = 0$), we only observe the one-sided negative outcome, so the source $\Omega_{y=1,s=0}$ is missing; 2) *missing subgroup*: we have training data for males with positive and negative outcomes, but do not have labelled data for females, i.e.both $\Omega_{y=1,s=0}$ and $\Omega_{y=0,s=0}$ are missing.

As before, `Ranking`, `k-means`, `No bal.` and `Perfect` refer to our method with different procedures for constructing (approximately) perfect bags. As baseline methods, we have `ERM` (standard empirical risk minimisation with balanced batches), `DRO` (Hashimoto et al., 2018), `gDRO` (Sagawa et al., 2019) and `ERM (LD)` which is the same model as `ERM`, but trained on the labelled deployment set, in addition to the training set.

In both settings, we observe the same order as for the other dataset in terms of accuracy: `Perfect` (with ground truth labels for balancing) achieves the highest performance, followed by `Ranking`, then `No bal.`, and finally `k-means`. However, for the *missing subgroup* setting, `Ranking` and `Perfect` are almost identical and the former performs better in terms of de-biasing metrics. This decreased reliance on balancing can be explained by the additional

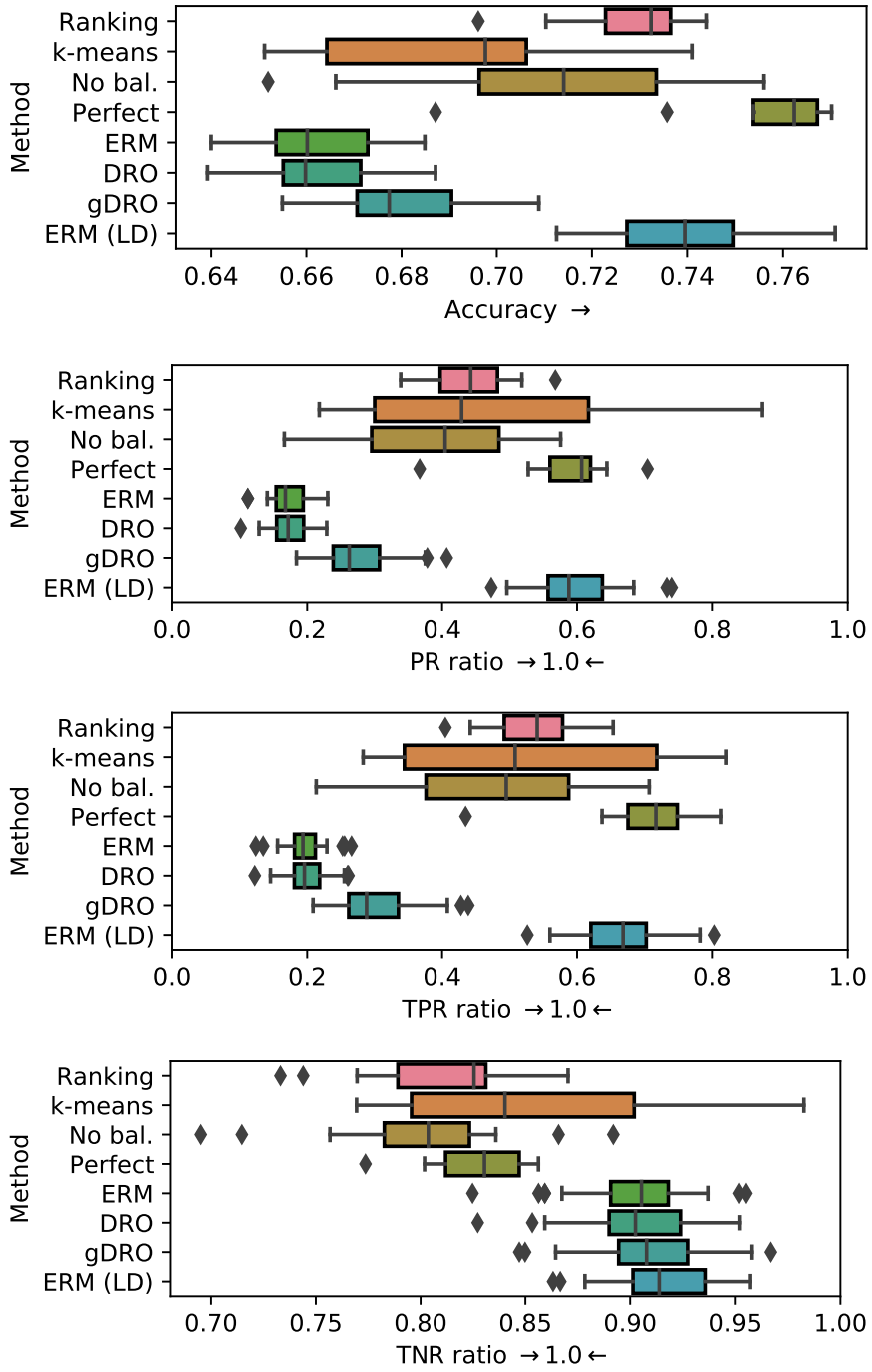


Figure 6.8: Results for the Adult Income dataset with *subgroup bias*, for the binary classification task of predicting whether an individual earns >$50,000 with a binary subgrouping based on *gender*. `ERM (LD)` refers to a model based on ERM (empirical risk minimisation), trained on a *l*abelled *d*eployment set; thus not suffering from bias in the training set. Top left: Accuracy. Top right: Positive rate ratio. Bottom left: True positive rate ratio. Bottom right: True negative rate ratio. For the `Ranking` clustering, the clustering accuracy was 69.7% ± 0.3%; for `K-means` it was 43% ± 3%.

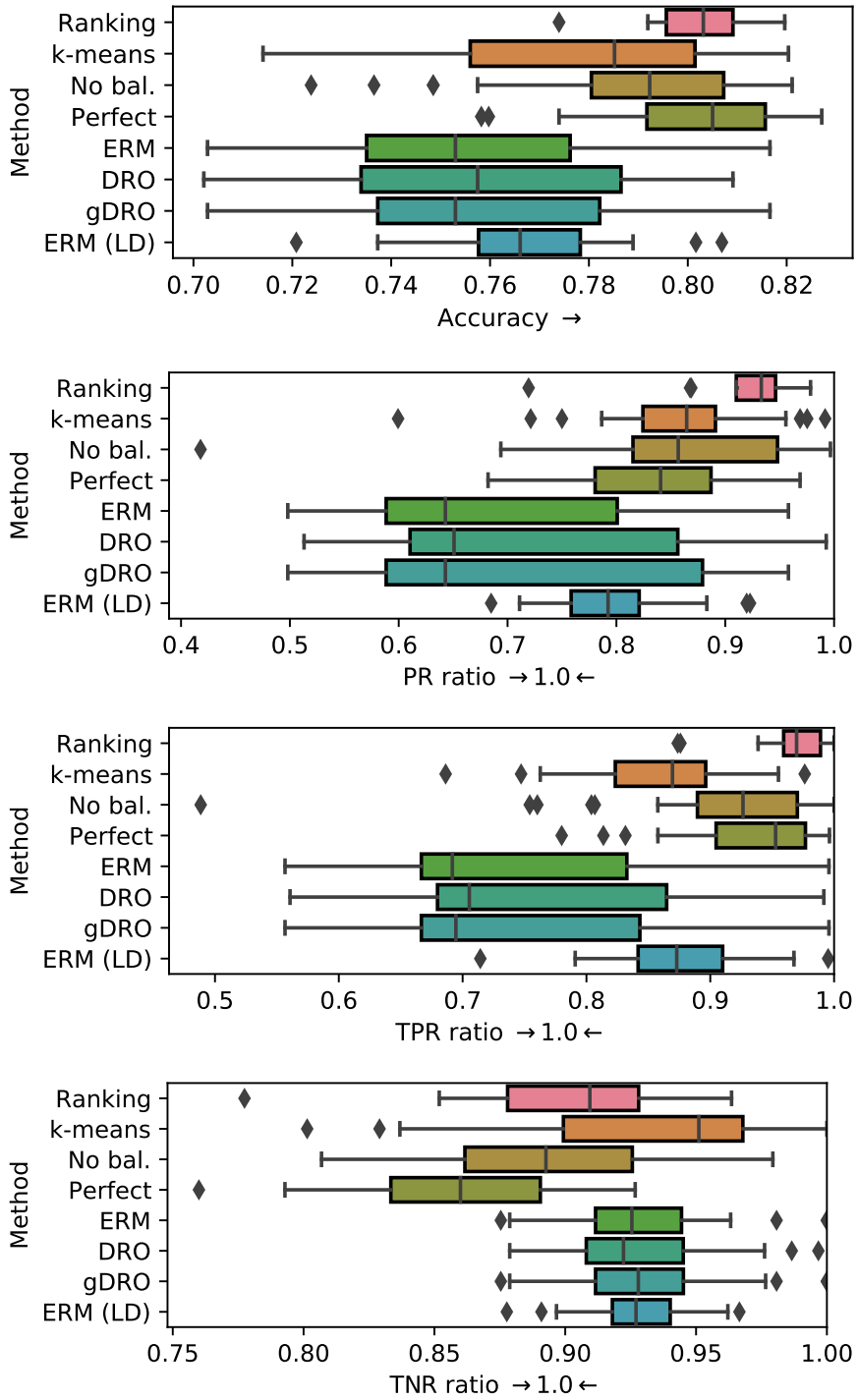


Figure 6.9: Results for the Adult Income dataset with a *missing subgroup*, for the binary classification task of predicting whether an individual earns >$50,000 with a binary subgrouping based on *gender*. `ERM (LD)` refers to a model based on ERM (empirical risk minimisation), trained on a *l*abelled *d*eployment set; thus not suffering from bias in the training set. Top left: Accuracy. Top right: Positive rate ratio. Bottom left: True positive rate ratio. Bottom right: True negative rate ratio. For the `Ranking` clustering, the clustering accuracy was 60.4% ± 0.8%; for `K-means` it was 44% ± 3%.

supervision that comes with having two sources missing instead of one - in order for the discriminator to distinguish between bags from the deployment set and bags from the training set, the former need only contain *one* of the two missing sources.

Generally, we observe a high variance in the results. This is not attributable to our method, however, with the baselines exhibiting the same behaviour, but rather to the fact that the Adult Income dataset is a very noisy dataset which, at the best of times, allows only about 85% accuracy to be attained (see also Agrawal et al., 2020). The problem is that samples vary widely in how informative they are. This, coupled with our artificially biasing the dataset to be even more biased (as *subgroup bias* and *missing subgroup*), makes the achievable performance very dependent on which samples the classifier gets to see, which varies according to the random seed used for the data set split.

6.7.2 *Dataset Construction*

COLOURED MNIST BIASING PARAMETERS. To simulate a real-word setting where the data, labelled or otherwise, is not naturally balanced, we bias the Coloured MNIST training and deployment sets by downsampling certain colour/digit combinations. The proportions of each such combination *retained* in the *subgroup bias* (in which we have one source missing from the training set) and *missing subgroup* (in which we have two sources missing from the training set) are enumerated in table 6.1 and 6.2, respectively. For the 3-digit-3-colour variant of the problem, no biasing is applied to either the deployment set or the training set (the missing combinations are specified in the caption accompanying figure 6.14); this variant was experimented with only under the subgroup-bias setting.

Table 6.1: Biasing parameters for the training (left) and deployment (right) sets of Coloured MNIST in the *subgroup bias* setting.

Combination	Proportion retained	
	training set	deployment set
(y = 2, s = purple)	1.0	0.7
(y = 2, s = green)	0.3	0.4
(y = 4, s = purple)	0.0	0.2
(y = 4, s = green)	1.0	1.0

ADULT INCOME. For the Adult Income dataset, we do not need to apply any synthetic biasing as the dataset naturally contains some bias wrt *s*. Thus, we instantiate the deployment set as just a random subset of the original dataset.

Table 6.2: Biasing parameters for the training (left) and deployment (right) sets of Coloured MNIST in the *missing subgroup* setting.

Combination	Proportion retained	
	training set	deployment set
(y = 2, s = purple)	0.0	0.7
(y = 2, s = green)	0.85	0.6
(y = 4, s = purple)	0.0	0.4
(y = 4, s = green)	1.0	1.0

Explicit balancing of the test set is needed to yield meaningful evaluation, however, namely in the penalising of biased classifiers, but need be taken in doing so. Balancing the test set such that

$$\begin{aligned} |\{x \in X | s = 0, y = 0\}| &= |\{x \in X | s = 1, y = 0\}| \\ \text{and } |\{x \in X | s = 0, y = 1\}| &= |\{x \in X | s = 1, y = 1\}| \end{aligned} \tag{6.9}$$

where for both target classes, $y = 0$ and $y = 1$, the proportions of the groups $s = 0$ and $s = 1$ are made to be the same, is intuitive, yet at the same time precludes sensible comparison of the accuracy/fairness trade-off of the different classifiers. Indeed, with the above conditions, a majority classifier (predicting all 1s or 0s) achieves comparable accuracy to the fairness-unaware baselines, while also yielding perfect fairness, by construction. This observation motivated us to devise an alternative scheme, where we balance the test set according to the following constraints

$$\begin{aligned} |\{x \in X | s = 0, y = 0\}| &= |\{x \in X | s = 0, y = 1\}| \\ = |\{x \in X | s = 1, y = 1\}| &= |\{x \in X | s = 1, y = 0\}| \,. \end{aligned} \tag{6.10}$$

That is, all subsets of $\mathcal{S} \times \mathcal{Y}$ are made to be equally sized. Under this new scheme the accuracy of the the majority classifier is 50% for the binary-classification task.

6.7.3 *Optimisation*

The hyperparameters and architectures for the Autoencoder (`AE`), Predictor and Discriminator subnetworks used for the experiments with all datasets are detailed in Table 6.3. All networks are trained using the Adam optimiser (Kingma and Ba, 2015).

For the Coloured MNIST and CelebA datasets, the baseline `CNN`, `DRO`, and `LfF` (in the case of the former) models use an architecture identical to that of

Table 6.3: Selected hyperparameters for experiments with Coloured MNIST, Adult and CelebA datasets.

	Coloured MNIST 2-dig SB / 2-dig MS / 3-dig SB	Adult	CelebA
Input size	$3 \times 32 \times 32$	61	$3 \times 64 \times 64$
	Autoencoder		
Levels	4	1	5
Level depth	2	1	2
Hidden units / level	[32, 64, 128, 256]	[61]	[32, 64, 128, 256, 512]
Activation	GELU	GELU	GELU
Downsampling op.	Strided Convs.	–	Strided Convs.
Reconstruction loss	MSE	Mixed[1]	MSE
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
	Clustering		
Batch size	256	1000	–
AE pre-training epochs	150	100	–
Clustering epochs	100	300	–
Self-supervised loss	Cosine + BCE	Cosine + BCE	–
U (for ranking statistics)	5	3	–
	Distribution Matching		
Batch size	1/32/14	64	32
Bag size	256/8/18	32	8
Training iterations	8k/8k/20k	5k	15k
Encoding (z) size[2]	128	35	128
Binarised z_s	✗ / ✓ / ✓	✗	✗
y-predictor weight (λ_1)	1	0	1
s-predictor weight (λ_2)	1	0	0
Adversarial weight (λ_3)	1×10^{-3}	1	1
Stopgradient $\left(\nabla_\theta h_\psi(f_\theta(X^{dep})) = 0\right)$	✗	✓	✗
	Predictors		
Learning rate	3×10^{-4}	1×10^{-3} 1×10^{-3}	
	Discriminator		
Attention mechanism[3]	Gated	Gated	Gated
Hidden units pre-aggregation	[256, 256]	[32]	[256, 256]
Hidden units post-aggregation	[256, 256]	–	[256, 256]
Embedding dim (for attention)	32	128	128
Activation	GELU	GELU	GELU
Learning rate	3×10^{-4}	1×10^{-3}	1×10^{-3}
Updates / AE update	1	3	1

[1] Cross-entropy is used for categorical features, MSE for continuous features.

[2] $|z|$ denotes the combined size of z_s and z_y, with the former occupying $\lceil \log_2(\mathcal{S}) \rceil$ dimensions, the latter the remaining dimensions.

[3] The attention mechanism used for computing the sample-weights within a bag. *Gated* refers to gated attention proposed by Ilse et al., 2018.

the encoder with two exceptions: 1) max-pooling being used for spatial down-sampling instead of strided convolutions; 2) the final convolutional layer is followed by a global average pooling layer followed by a fully-connected classification layer. For evaluating our method, we simply train a linear classifier on top of z_y; this is sufficient due to linear-separability being enforced during training by the y-predictor. For the Adult Income dataset, we use an MLP made up of a single hidden layer – 35 units in size – followed by a SELU activation (Klambauer et al., 2017), as both both the downstream classifier for our method, and as the network architecture of the baselines. All baselines and downstream classifiers alike were trained for 60 epochs with a learning rate of 1×10^{-3} and a batch size of 256.

Since, by design, we do not have labels for all subgroups the model will be tested on, and bias against these missing subgroups is what we aim to avoid, properly validating, and thus conducting hyperparameter selection for models generally, is not straightforward. We can use estimates of the mutual information between the learned-representation and s and y (which we wish to minimise w.r.t. to the former, maximise w.r.t. the latter) to guide the process, though optimising the model w.r.t. these metrics obtained from only the training set does not guarantee generalisation to the missing subgroups. We can, however, additionally measure the entropy of the predictions on the encoded test set and seek to maximise it across all samples, or alternatively train a discriminator of the same kind used for distribution matching as a measure the shift in the latent space between datasets. We use the latter approach (considering, the learned distance between subspace distributions, accuracy, and reconstruction loss) to inform an extensive grid-search over the hyperparameter space of our model.

For the `DRO` baseline, we allowed access to the labels of the test set for the purpose of hyperparameters selection, performing a grid-search over multiple splits to avoid overfitting to any particular instantiation. Specifically, the threshold (η) parameter for `DRO` was determined by a grid-search over the space $\{0.01, 0.1, 0.3, 1.0\}$. The same procedure was carried out for selecting the model capacity constant (C) of the related `gDRO` baseline.

In addition to the losses stated in the distribution matching objective, $\mathscr{L}$, in the main text, we also regularise the encoder by the ℓ^2 norm of its embedding, multiplied by a small pre-factor, finding this to work better than more complex regularisation methods, such as spectral normalisation (Miyato et al., 2018), for stabilising training.

Figure 6.10: Example sample-wise attention maps for bags of CelebA (left) and Coloured MNIST (right) images sampled from a balanced deployment set. The training set is biased according to the SB setting where for CelebA "smiling females" constitute the missing source and for Coloured MNIST purple fours constitute the missing source. The attention weights are used during the discriminator's aggregation step to compute a weighted sum over the bag. The attention-weight assigned to each sample is proportional to the lightness of its frame, with black signifying a weight of 0, white a weight of 1. Those samples belonging to the missing subgroup are assigned the highest weight as they signal from which dataset (training vs. deployment) the bag containing them was drawn from.

6.7.4 *Visualisations of results*

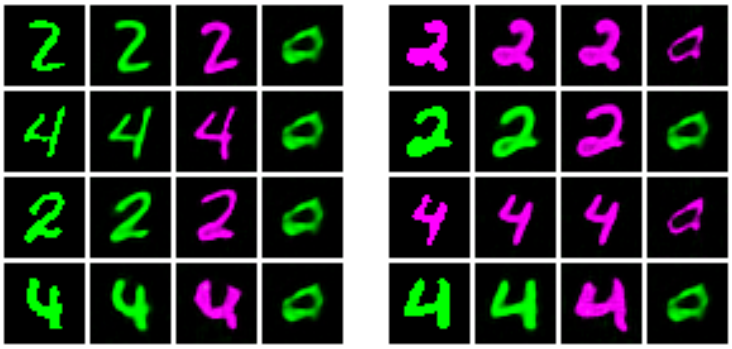
Figures 6.10 and 6.11 show some additional visualisations for our results. For details, see the captions.

6.7.5 *Code*

The code will be published at the following URL: `https://github.com/predictive-analytics-lab/fair-dist-matching`. Instructions on how to run them can be found in the `README.md`.

6.7.6 *Additional metrics*

Figures 6.12, 6.13, and 6.15 show the true positive rate (TPR) ratio and the true negative rate (TNR) ratio as additional metrics for Coloured MNIST (2 digits) and CelebA. These are computed as the ratio of TPR (or TNR) on subgroup $s = 0$ over the TPR (or TNR) on subgroup $s = 1$; if this gives a number greater than 1, the inverse is taken. Similarly to the PR ratio reported in the main paper, these ratios give an indication of how much the prediction of the classifier depends on the subgroup label s.



(a) Different reconstructions on the training set. Corresponding to: original, full reconstruction, reconstruction of z_y, reconstruction of z_s.

(b) Different reconstructions on the deployment set. Corresponding to: original, full reconstruction, reconstruction of z_y, reconstruction of z_s.

Figure 6.11: Visualisation of our method's solutions for the Coloured MNIST dataset, with purple as the missing subgroup. In each of the subfigures 6.11a and 6.11b: Column 1 shows the original images from x from the respective set. Column 2 shows plain reconstructions generated from $x_{recon} = g(f_y(x), f_s(x))$. Column 3 shows reconstruction with zeroed-out z_s: $g(f_y(x), 0)$, which effectively visualises z_y. Column 4 shows the result of an analogous process where z_y was zeroed out instead.

Figure 6.14 shows metrics specific to multi-valued s (i. e., non-binary s). We report the minimum (i.e. farthest away from 1) of the pairwise ratios (PR/TPR/TNR ratio min) as well as the largest difference between the raw values (PR/TPR/TNR diff max) . Additionally, we compute the Hirschfeld-Gebelein-Rényi (HGR) maximal correlation (Rényi, 1959) between S and Y, serving as a measure of dependence defined between two variables with arbitrary support.

6.7.7 *Clustering with an incorrect number of clusters*

We also investigate what happens when the number of clusters is set incorrectly. For 2-digit Coloured MNIST, we expect 4 clusters, corresponding to the 4 possible combinations of the binary class label y and the binary subgroup label s. However, there might be circumstances where the correct number of clusters is not known; how does the batch balancing work in this case? We run experiments with the number of clusters set to 6 and to 8, while otherwise not changing any part of the method. It should be noted that this is a very naïve way of dealing with an unknown number of clusters. There are methods specifically designed for identifying the right number of clusters (Hamerly and Elkan, 2003; Chazal et al., 2013), and that is what would be used if this situation came up in practice.

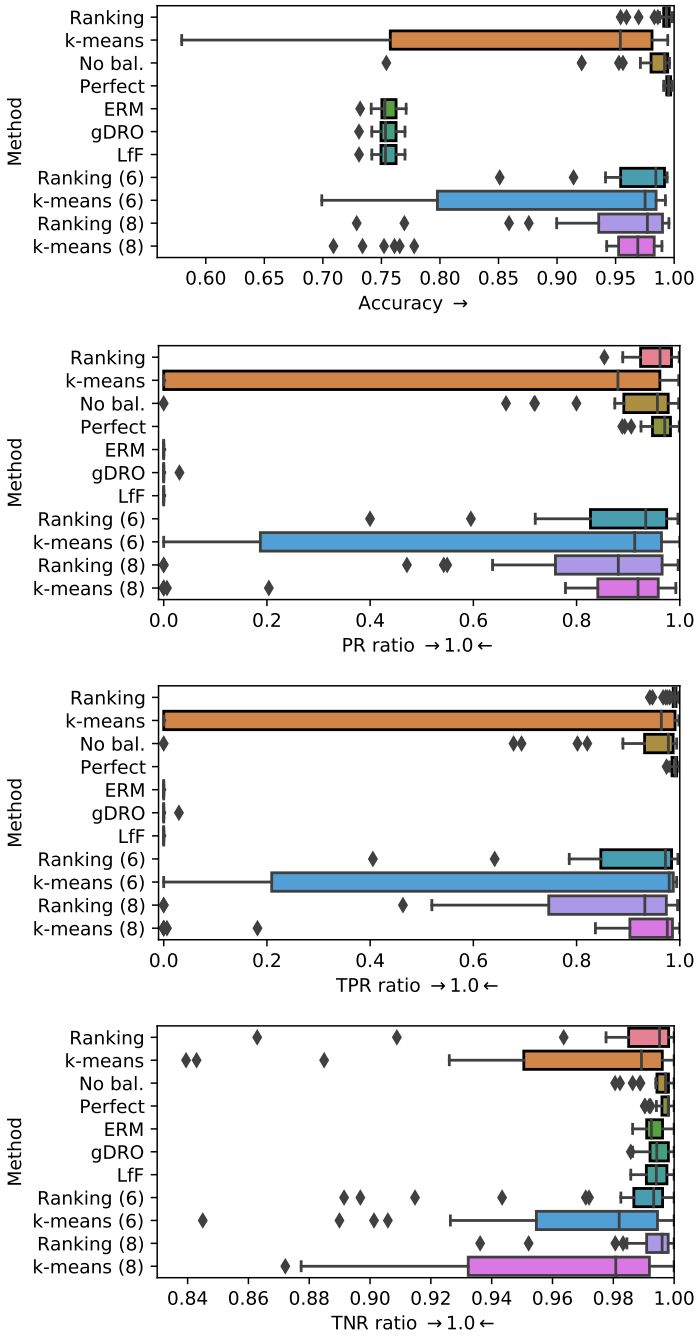


Figure 6.12: Results from 30 repeats for the Coloured MNIST dataset with two digits, 2 and 4, with *subgroup bias* for the colour 'purple': for purple, only the digit class '2' is present. Top left: Accuracy. Top right: Positive rate ratio. Bottom left: True positive rate ratio. Bottom right: True negative rate ratio. For the `Ranking` clustering, the clustering accuracy was 96% ± 6%; for `K-means` it was 64% ± 10%. For an explanation of `Ranking (8)` and `K-means (8)` see section 6.7.7.

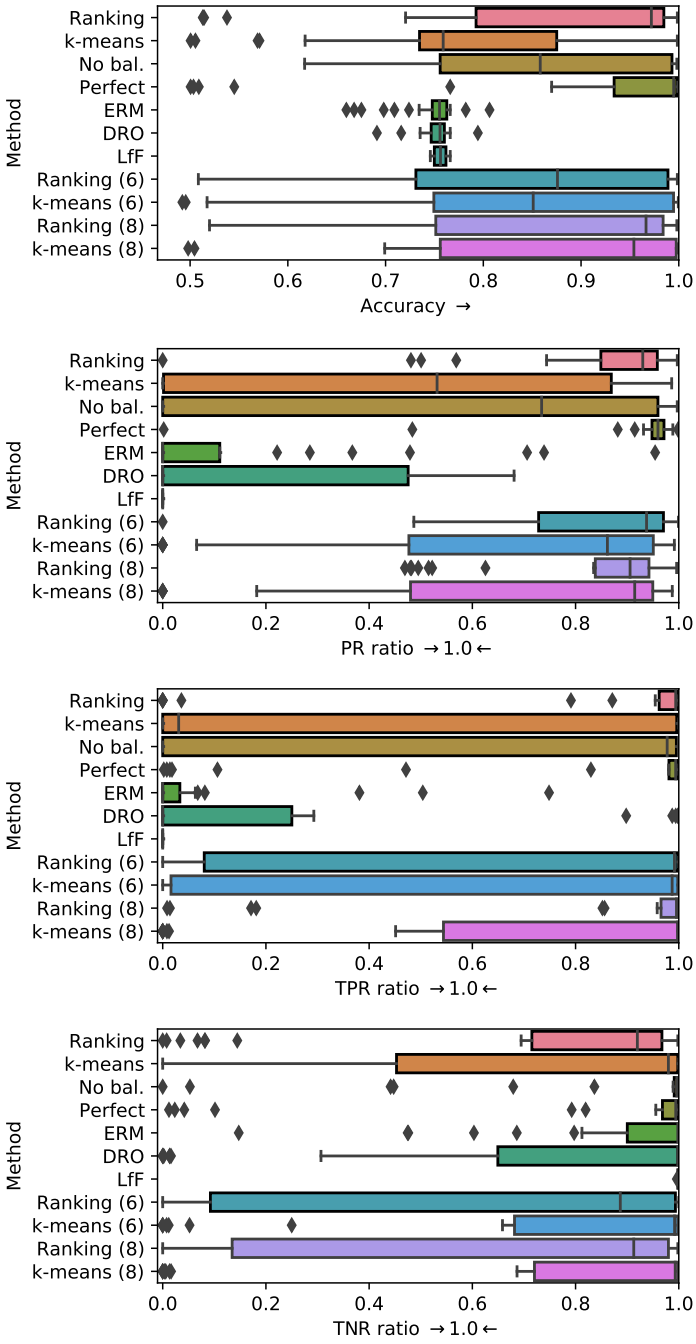


Figure 6.13: Results from 30 repeats for the Coloured MNIST dataset with two digits, 2 and 4, with a *missing subgroup*: the training dataset only has green digits. TOP LEFT: Accuracy. TOP RIGHT: Positive rate ratio. BOTTOM LEFT: True positive rate ratio. BOTTOM RIGHT: True negative rate ratio. For the `Ranking` clustering, the clustering accuracy was 88% ± 5%; for `K-means` it was 72% ± 16%. For an explanation of `Ranking (8)` and `K-means (8)` see section 6.7.7.

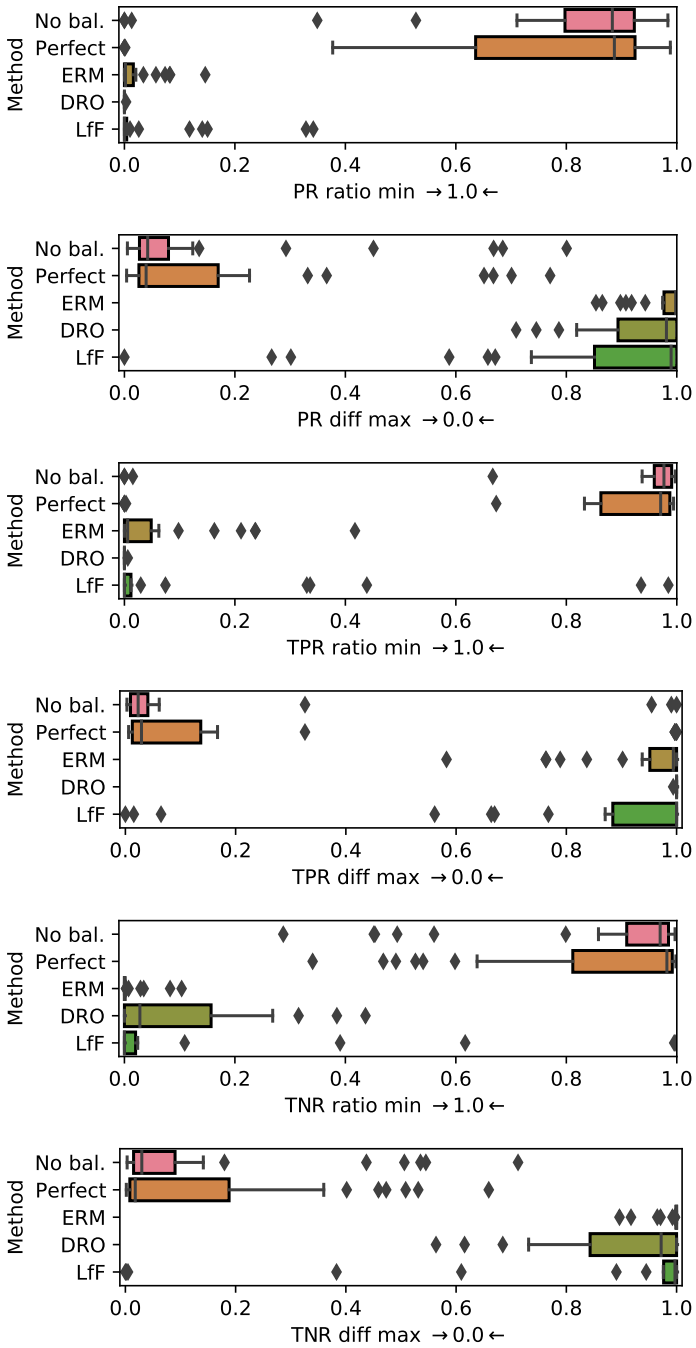


Figure 6.14: Results from 30 repeats for the Coloured MNIST dataset with three digits: '2', '4' and '6'. Four combinations of digit and colour are missing: green 2's, blue 2's, blue 4's and green 6's. First row, left: minimum of all positive rate ratios. First row, right: maximum of all positive rate differences. Second row, left: minimum of all true positive rate ratios. Second row, right: maximum of all true positive rate differences. Third row, left: minimum of all true negative rate ratios. Third row, right: maximum of all true negative rate differences.

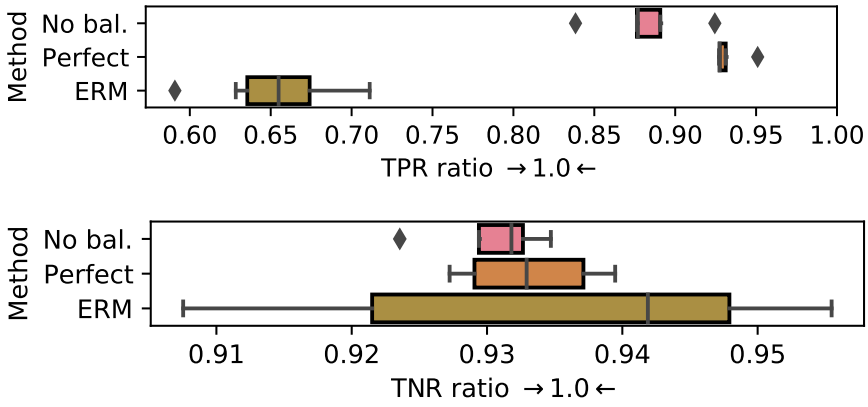


Figure 6.15: Results from 10 repeats for the CelebA dataset with the *subgroup bias* setting. The task is to predict "smiling" vs "non-smiling" and the subgroups are based on gender. The subgroup "female" is missing samples for the "smiling" class. LEFT: True positive rate ratio. RIGHT: True negative rate ratio.

The results can be found in figures 6.12 and 6.13. Bags and batches are constructed by drawing an equal number of samples from each cluster. Unsurprisingly, the method performs worse than with the correct number of clusters. When investigating how the clustering methods deal with the larger number of clusters, we found that it is predominantly those samples that do not appear in the training set which get spread out among the additional clusters. This is most likely due to the fact that the clustering is semi-supervised, with those clusters that occur in the training set having supervision. The overall effect is that the samples which are not appearing in the training set are overrepresented in the drawn bags, which means it is easier for the adversary to identify where the bags came from, and the encoder cannot properly learn to produce an invariant encoding.

Part III

CONCLUSION

7 DISCUSSION AND FUTURE WORK

In this thesis, I have presented three approaches for dealing with dataset bias. The first one deals with a form of label bias and the other two with forms of sampling bias; in both cases the bias is linked to a special attribute s of the data. With the first approach, the user specifies target rates that can be used to tune the method to the desired outcome. The method is flexible and easy to integrate with existing algorithms. The second approach uses a partially-labelled representative set to learn an interpretable and invariant representation. The user can directly observe in what way the data was changed to make it invariant to the spurious correlation that is present in the training data. The third approach in many ways builds upon the second one; a set similar to the representative set is needed but there is no requirement for labels nor for balance of subgroups. However, in the training set, the relationship between the class label y and subgroups s may not be as close to a one-to-one mapping of y and s as in the second approach. From these simple starting points, an invariant representation is learned that can be used to train unbiased classifiers.

The remainder of this chapter is split into three parts: In section 7.1, I discuss some of the limitations of the presented methods to keep in mind, and the goals that can be achieved by applying the methods. Section 7.2 is about ways in which the work could be extended, and section 7.3 discusses the broader perspective of this topic, including a wider discussion of the future of the field.

7.1 Limitations and intended use

The presented work by no means covers *all* possible biases, but it contributes to a growing literature that tries to tackle this problem. One could ask if there is one method that is able to cover all possible dataset biases, but I think there is a strong argument to be made that no general method can exist, because it is, e. g., not possible to describe in general what is spurious information and what is relevant information. Nevertheless, finding methods that are more generally applicable is a worthy goal. One immediate avenue for future work is the combination of label bias correction and sampling bias correction.

For example, it might be possible to leverage an unlabelled context set for correcting label bias as well.

It also should be mentioned here that compared to training straightforward, bias-unaware classifiers, the sampling-bias-targeting methods (i. e., the second and the third method) incur additional costs in terms of computing resources, training time and human labour. The label-bias-targeting method (i. e., the first one) only impacts the computation of the loss and does not require additional training data, so it has a negligible impact on training cost. The second method is the most costly as the training of INNs requires a large amount of memory and compute because INNs only *transform* the (very high-dimensional) inputs and never perform anything like the lossy compression that other neural network do. Furthermore, the method requires additional data with s labels. Still, if the additional training cost enables the use of data that was previously unusable due to strong spurious correlations, then it can still be worthwhile, and it is also important to keep in mind that a model can be trained once and then used millions of times. This has to be decided on a case-by-case basis, and unfortunately means that these de-biasing methods will likely not be use prophylactically, but only if there is an explicit expectation of harm otherwise.

In any case, correcting dataset bias remains a challenging topic and one that is increasingly relevant to today's machine learning applications. Any cutting-edge ML system will have to deal with imperfect data, especially if the collected data is human-related. The possible effects of these imperfections in the data are certainly highly undesirable: a photo-tagging service might only work for a certain kind of person; a speech recognition system might only work for a certain kind of dialect. If, in these situations, sufficient representative (but unlabelled) data is available, then the methods presented here can be used to try and correct the problem.

Now, one possible objection here is: if those datasets are of such poor quality, then maybe we should not train any ML model on these and should not use them to make automated decisions. While this question is mostly beyond the scope of this document, let me offer some thoughts on this: It is true that even after the application of de-biasing techniques, the resulting models still should not be fully trusted, but they can still ease the burden of tedious manual labour; similar to an email spam filter which is not perfect, but still very useful. Or, put another way, it is always important to check what the realistic alternative is; we should not compare a model to a non-existing perfect ideal, but to the actual solution that would be used instead. One could imagine a hybrid approach where an automated system makes preliminary decisions, but random samples are reviewed by humans and decisions can

always be challenged. Ideally, the model itself would tell us about decisions it is uncertain about.

Moreover, two of the three methods presented in this thesis require access to (unlabelled) *unbiased* data, which is used during the training process. The remaining method requires summary statistics of the unbiased target dataset. So, the criticism that we are only learning from biased data does not apply here. This should allow us to be more confident in the predictions produced by those methods, because the methods learn from additional, unbiased information.

The last limitation to mention is that throughout the thesis, the assumption is made that all relevant 'demographic groups' (or 'environments' or 'subgroups') are known, or at least that it is known how *many* groups there are. This goes back to the argument above that no *completely general* method can exist to solve dataset bias; there has to be *some* inductive bias in order to be able to learn anything. There have been attempts in the literature to address this, like Hashimoto et al. (2018) (see section 2.5), which only requires knowledge of a lower bound on the size of groups in the given data. Such an assumption could potentially replace the assumption of knowing the *number* of groups (as we assume for the third method presented). The other direction to go in is to give up on groups altogether and try to enforce *individual fairness*. However, this has its own set of problems; the foremost of which is requiring a sensible distance function. Given these issues, it is likely not possible to remove all bias everywhere (though that should not discourage us from trying to).

A topic that was left out of the thesis is *data augmentation*, which refers to the practice of modifying copies of the samples in the training set with pre-defined transformations, which do not change the "semantic content", such that more samples are available for training. For image data, typical transformations are rotations, crops, mirroring, Gaussian noise, and slight colour modifications. If chosen right, data augmentation alone can solve the problems presented in this thesis: grey-scaling is a transformation that is sometimes used for image data, and works quite well on the Coloured MNIST problem (though not perfectly well, because even as shades of grey, the colours are distinguishable). However, this only works because the spurious feature is so simply here (after all, Coloured MNIST is mostly intended to be a simple toy dataset). Grey-scaling would not help with real-world datasets like CelebA. Furthermore, even if a simple transformation would be able to remove the spurious feature, knowing *which* transformation will accomplish this is not always as obvious as with Coloured MNIST, and may require special domain knowledge (like signal processing for audio data). Thus, the

goal for the presented methods here was to make them applicable to any kind of data, without requiring knowledge of the specific details of how the bias manifests itself in the data, and without requiring knowledge of which data augmentations can be applied. However, this is not a recommendation *against* data augmentations; if you have reason to believe that augmentations will help, there is no reason not to use them.

7.2 Potential extensions

In general, it would be desirable to have more theoretical bounds on performance. This would likely involve specifying the requirements for the algorithms more precisely. The method based on target labels would furthermore benefit from better-calibrated probability outputs. Gaussian process (GP) classifiers show generally good calibration, but cannot be easily applied in domains where deep neural networks are used. On the other hand, neural networks are known to be overconfident (especially when using ReLU activations; Hein et al., 2019), so to apply the proposed method to images directly is not straightforward. The other two proposed methods would benefit from improved adversarial training procedures. Training these models is not straightforward as the losses have to be balanced and in some cases, the update schedule needs to be changed. Both the calibration of neural networks and the stability of adversarial training are issues that are widely recognised in the ML community, so there is hope that progress will be made on these.

Another potential extension of this work is to extend it to other modalities. The experiments were all performed on either tabular or image data — the reason predominantly being ease of visualisation — so working with audio (especially speech) or text could present new, interesting challenges.

Furthermore, as it currently stands, the ML practitioner has to know the bias in the data very well in order to choose a method to correct it. It would be desirable to have a simple algorithm for deciding which method to use. A related limitation is that the dataset bias considered in this thesis is assumed to be linked to a special attribute s. While s can be very high-dimensional and is not limited to, say, binary attributes, this nevertheless represents a restriction that excludes large areas of dataset biases.

A potential – very ambitious – extension would be to try to learn to automatically detect biases: i. e. an algorithm that can recognise biases without asking a human operator. The question is then, where can an ML algorithm learn about all possible biases? Given the complexity of human values (Yudkowsky, 2011), it will not be possible to compress knowledge of all

biases down to a compact representation. Essentially, the only way is to learn them all individually, one by one. This could potentially be done with models like GPT-3 (Brown et al., 2020), which are trained on enormous amounts of text data, such that they conceivably have learned a lot about what human writers consider to be biases. There is then perhaps a way to extract what such a model knows about biases (it could be as easy as a text prompt which asks "is it okay to discriminate based on gender?"), which would allow us to check whether those biases exist in our data. Of course, this assumes that humans agree amongst themselves what biases are – otherwise the model might be very unreliable – which seems like quite a risky assumption to make. Furthermore, prevalent ethics have change over the course of human history, and are likely to change in the future as well. The ideal solution would be to predict which ethics humans will adopt in the future, after they thought about it even more, which is the goal of *coherent extrapolated volition* (Tarleton, 2010), but has large technical hurdles. There are also other potential problems with unsupervised models like GPT-3; some of which are discussed below.

7.3 Broader perspective

An important issue is the communication between human and machine. The methods presented in this thesis have all strived to make it easy for human operators to understand what is going on: the method in chapter 4 is configured with simple statistics; the method in chapter 5 produces invariant images that can be inspected; and the method in chapter 6 has the same capability (though it is not part of the core functionality). However, this is only a beginning. It is still not easy to notice that a given dataset has problems, and *currently*, machines on their own cannot notice the problem (see above). Thus, it is important that machines get feedback early and often, to keep them aligned with human goals. Applied to the problem of dataset bias, this could mean visualising correlations in the dataset, or routinely producing invariant representations to show what the network thinks it is meant to learn.

One area of machine learning that has recently seen increased interest is unsupervised learning, and the latter two chapters make use of it to some degree. The exciting promise of unsupervised learning in general is that labour-intensive labelling is not needed and so vast amounts of existing, unlabelled data can be put to good use. One could ask the question whether bias-correcting methods are still needed, with access to so much data. It could be that, while the data is certainly not perfect, there is so much of it

that the biased parts "cancel each other out". However, recent investigations into the GPT models (Radford et al., 2018, 2019; Brown et al., 2020) do not seem to support this (Khalifa et al., 2021). One reason for this might be the way these models are trained at the moment: they maximise the probability assigned to the next token. Thus, such a model has to account for the wide array of human opinions and assign a non-zero probability to all of them. So, when asked to summarise a text (Stiennon et al., 2020), GPT does not give the *best* summary; rather, it gives a summary that an average person might have written. However, with the help of a very high-quality labelled dataset (that was expensive to create), it was possible to finetune GPT to actually produce very good summaries. I suspect this pattern of learning the basics in an unsupervised fashion, and then finetuning with high-quality labels, will continue in the future. With these massive models, it is, more than ever, crucial to build tools to make the biases within the models transparent. Given the black-box nature of neural networks, this represents a major challenge.

It was mentioned before that an ML model should be judged by how useful it is, how it stacks up to realistic alternatives, and not whether it is perfect. However, one has to be careful not to take this philosophy too far. Namely, one should avoid giving up and saying, "why should AI need to be fair if humans tend to be biased anyhow?" The faults of humans has little to do with the question of what we expect of machines. If one of our machines will affect many of our fellow humans, then I think we would not want it to be harmful. Should we make the machine artificially biased so that humans do not need to feel too guilty about their own biases? I believe this to be folly. We should always strive to do the best we can and that means making machines as unbiased as possible.

BIBLIOGRAPHY

Adel, Tameem, Zoubin Ghahramani and Adrian Weller (2018). 'Discovering Interpretable Representations for Both Deep Generative and Discriminative Models'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 50–59.

Agarwal, Alekh, Alina Beygelzimer, Miroslav Dudík, John Langford and Hanna M. Wallach (2018). 'A Reductions Approach to Fair Classification'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 60–69.

Agrawal, Ashrya et al. (2020). 'Debiasing classifiers: is reality at variance with expectation?' In: *arXiv preprint arXiv:2011.02407*.

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

Ardizzone, Lynton, Carsten Lüth, Jakob Kruse, Carsten Rother and Ullrich Köthe (2019). 'Guided Image Generation with Conditional Invertible Neural Networks'. In: *arXiv preprint arXiv:1907.02392*.

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani and David Lopez-Paz (2019). 'Invariant risk minimization'. In: *arXiv preprint arXiv: 1907.02893*.

Barocas, Solon, Moritz Hardt and Arvind Narayanan (2019). *Fairness and Machine Learning*. http://www.fairmlbook.org.

Barocas, Solon and Andrew D. Selbst (2016). 'Big Data's Disparate Impact'. In: *California Law Review* 104, pp. 671–732.

Barredo Arrieta, Alejandro et al. (2020). 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI'. In: *Information Fusion* 58, pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012.

Bengio, Yoshua, Nicholas Léonard and Aaron Courville (2013). 'Estimating or propagating gradients through stochastic neurons for conditional computation'. In: *arXiv preprint arXiv:1308.3432*.

Beutel, Alex, Jilin Chen, Zhe Zhao and Ed H. Chi (2017). 'Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations'. In: *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*.

Bickel, Peter J, Eugene A Hammel and J William O'Connell (1975). 'Sex bias in graduate admissions: Data from Berkeley'. In: *Science* 187.4175, pp. 398–404.

Blum, Avrim and Kevin Stangl (2020). 'Recovering from Biased Data: Can Fairness Constraints Improve Accuracy?' In: *1st Symposium on Foundations of Responsible Computing (FORC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Bonilla, Edwin V, Karl Krauth and Amir Dezfouli (2016). 'Generic Inference in Latent Gaussian Process Models'. In: *arXiv preprint arXiv:1609.00577*.

Brown, Tom B. et al. (2020). 'Language Models are Few-Shot Learners'. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Bucher, Maxime, Tuan-Hung Vu, Matthieu Cord and Patrick Pérez (2019). 'Zero-Shot Semantic Segmentation'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 466–477.

Calders, Toon, Faisal Kamiran and Mykola Pechenizkiy (2009). 'Building classifiers with independency constraints'. In: *IEEE International Conference on Data Mining Workshops*. IEEE, pp. 13–18.

Calders, Toon and Sicco Verwer (2010). 'Three naive Bayes approaches for discrimination-free classification'. In: *Data Mining and Knowledge Discovery* 21.2, pp. 277–292.

Chapelle, Olivier, Bernhard Schölkopf and Alexander Zien (2006). 'Introduction to Semi-Supervised Learning'. In: *Semi-Supervised Learning*. The MIT Press, pp. 1–12.

Chazal, Frédéric, Leonidas J. Guibas, Steve Y. Oudot and Primoz Skraba (2013). 'Persistence-Based Clustering in Riemannian Manifolds'. In: *J. ACM* 60.6. DOI: 10.1145/2535927.

Chiappa, Silvia (2019). 'Path-Specific Counterfactual Fairness'. In: *AAAI Conference on Artificial Intelligence*, pp. 7801–7808. DOI: 10.1609/aaai.v33i01.33017801.

Chouldechova, Alexandra (2017). 'Fair prediction with disparate impact: A study of bias in recidivism prediction instruments'. In: *Big data* 5.2, pp. 153–163.

Clanuwat, Tarin, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto and David Ha (2018). 'Deep learning for classical japanese literature'. In: *arXiv preprint arXiv:1812.01718*.

Cohen, Gregory, Saeed Afshar, Jonathan Tapson and Andre Van Schaik (2017). 'EMNIST: Extending MNIST to handwritten letters'. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 2921–2926.

Corbett-Davies, Sam, Emma Pierson, Avi Feller, Sharad Goel and Aziz Huq (2017). 'Algorithmic Decision Making and the Cost of Fairness'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 797–806. DOI: 10.1145/3097983.3098095.

Cotter, Andrew, Heinrich Jiang, Serena Wang, Taman Narayan, Maya R. Gupta, Seungil You and Karthik Sridharan (2018). 'Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals'. In: *arXiv preprint arXiv:1809.04198*.

Creager, Elliot, Jörn-Henrik Jacobsen and Richard Zemel (2020a). 'Environment Inference for Invariant Learning'. In: *ICML Workshop on Uncertainty and Robustness*.

Creager, Elliot, David Madras, Jörn-Henrik Jacobsen, Marissa A. Weis, Kevin Swersky, Toniann Pitassi and Richard S. Zemel (2019). 'Flexibly Fair Representation Learning by Disentanglement'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 1436–1445.

Creager, Elliot, David Madras, Toniann Pitassi and Richard S. Zemel (2020b). 'Causal Modeling for Fairness In Dynamical Systems'. In: *International Conference on Machine Learning (ICML)*. Vol. 119. Proceedings of Machine Learning Research, pp. 2185–2195.

DeDeo, Simon (2014). 'Wrong side of the tracks: Big Data and Protected Categories'. In: *arXiv preprint arXiv:1412.4643*.

Dheeru, Dua and Efi Karra Taniskidou (2017). *UCI Machine Learning Repository*.

Dimitrakakis, Christos, Yang Liu, David C. Parkes and Goran Radanovic (2019). 'Bayesian Fairness'. In: *AAAI Conference on Artificial Intelligence*, pp. 509–516. DOI: 10.1609/aaai.v33i01.3301509.

Dinh, Laurent, David Krueger and Yoshua Bengio (2014). 'NICE: Non-linear Independent Components Estimation'. In: *International Conference on Learning Representations (ICLR)*.

Donini, Michele, Luca Oneto, Shai Ben-David, John Shawe-Taylor and Massimiliano Pontil (2018). 'Empirical Risk Minimization Under Fairness Constraints'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2796–2806.

Dunnmon, Jared A, Darvin Yi, Curtis P Langlotz, Christopher Ré, Daniel L Rubin and Matthew P Lungren (2019). 'Assessment of convolutional neural networks for automated classification of chest radiographs'. In: *Radiology* 290.2, pp. 537–544.

Durugkar, Ishan P., Ian Gemp and Sridhar Mahadevan (2017). 'Generative Multi-Adversarial Networks'. In: *International Conference on Learning Representations (ICLR)*.

Dwork, Cynthia, Moritz Hardt, Toniann Pitassi, Omer Reingold and Richard Zemel (2012). 'Fairness through awareness'. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. ACM, pp. 214–226.

Edwards, Harrison and Amos J. Storkey (2016). 'Censoring Representations with an Adversary'. In: *International Conference on Learning Representations (ICLR)*.

Feldman, Michael, Sorelle A. Friedler, John Moeller, Carlos Scheidegger and Suresh Venkatasubramanian (2015). 'Certifying and Removing Disparate Impact'. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 259–268. DOI: 10.1145/2783258.2783311.

Feng, Rui, Yang Yang, Yuehan Lyu, Chenhao Tan, Yizhou Sun and Chunping Wang (2019). 'Learning fair representations via an adversarial framework'. In: *arXiv preprint arXiv:1904.13341*.

ForeverData.org (2015). *Heritage Health Prize Contest Data*.

Forman, George (2005). 'Counting positives accurately despite inaccurate classification'. In: *European Conference on Machine Learning*. Springer, pp. 564–575.

Friedler, Sorelle A, Carlos Scheidegger and Suresh Venkatasubramanian (2016). 'On the (im) possibility of fairness'. In: *arXiv: 1609.07236*.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand and Victor Lempitsky (2016). 'Domain-adversarial training of Neural Networks'. In: *Journal of Machine Learning Research* 17.1, pp. 2096–2030.

Gansbeke, Wouter Van, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans and Luc Van Gool (2020). 'SCAN: Learning to Classify Images Without Labels'. In: *European Conference on Computer Vision (ECCV)*, pp. 268–285.

Gardner, Jacob R., Geoff Pleiss, Kilian Q. Weinberger, David Bindel and Andrew Gordon Wilson (2018). 'GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7587–7597.

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann and Wieland Brendel (2019). 'ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness'. In: *International Conference on Learning Representations (ICLR)*.

Goh, Gabriel, Andrew Cotter, Maya R. Gupta and Michael P. Friedlander (2016). 'Satisfying Real-world Goals with Dataset Constraints'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2415–2423.

Hamerly, Greg and Charles Elkan (2003). 'Learning the k in k-means'. In: *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pp. 281–288.

Han, Kai, Sylvestre-Alvise Rebuffi, Sébastien Ehrhardt, Andrea Vedaldi and Andrew Zisserman (2020). 'Automatically Discovering and Learning New Visual Categories with Ranking Statistics'. In: *International Conference on Learning Representations (ICLR)*.

Hardt, Moritz, Eric Price and Nati Srebro (2016). 'Equality of Opportunity in Supervised Learning'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3315–3323.

Hashimoto, Tatsunori B., Megha Srivastava, Hongseok Namkoong and Percy Liang (2018). 'Fairness Without Demographics in Repeated Loss Minimization'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 1934–1943.

Hébert-Johnson, Úrsula, Michael P. Kim, Omer Reingold and Guy N. Rothblum (2018). 'Multicalibration: Calibration for the (Computationally-Identifiable) Masses'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 1944–1953.

Hein, Matthias, Maksym Andriushchenko and Julian Bitterwolf (2019). 'Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–50. DOI: 10.1109/CVPR.2019.00013.

Higgins, Irina, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende and Alexander Lerchner (2018). 'Towards a Definition of Disentangled Representations'. In: *arXiv preprint arXiv:1812.02230*.

Higgins, Irina, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed and Alexander Lerchner (2017). 'beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework'. In: *International Conference on Learning Representations (ICLR)*.

High-Level Expert Group on Artificial Intelligence (Apr. 2019). *Ethics Guidelines for Trustworthy AI*.

Holstein, Kenneth, Jennifer Wortman Vaughan, Hal Daumé III, Miroslav Dudík and Hanna M. Wallach (2019). 'Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?' In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, p. 600. DOI: 10.1145/3290605.3300830.

Hurley, Mikella and Julius Adebayo (2017). 'Credit scoring in the era of big data'. In: *Yale Journal of Law and Technology* 18, pp. 148–216.

Ilse, Maximilian, Jakub M. Tomczak and Max Welling (2018). 'Attention-based Deep Multiple Instance Learning'. In: *International Conference on*

Machine Learning (ICML). Vol. 80. Proceedings of Machine Learning Research, pp. 2132–2141.

Ioffe, Sergey and Christian Szegedy (2015). 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: *International Conference on Machine Learning (ICML)*. Vol. 37. JMLR Workshop and Conference Proceedings, pp. 448–456.

Jacobsen, Jörn-Henrik, Jens Behrmann, Richard S. Zemel and Matthias Bethge (2019). 'Excessive Invariance Causes Adversarial Vulnerability'. In: *International Conference on Learning Representations (ICLR)*.

Jacobsen, Jörn-Henrik, Arnold W. M. Smeulders and Edouard Oyallon (2018). 'i-RevNet: Deep Invertible Networks'. In: *International Conference on Learning Representations (ICLR)*.

Jaiswal, Ayush, Rex Yue Wu, Wael Abd-Almageed and Prem Natarajan (2018). 'Unsupervised Adversarial Invariance'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5097–5107.

Jaiswal, Ayush, Rex Yue Wu, Wael AbdAlmageed and Premkumar Natarajan (2019). 'Unified Adversarial Invariance'. In: *arXiv preprint arXiv:1905.03629*.

Jiang, Heinrich and Ofir Nachum (2020). 'Identifying and Correcting Label Bias in Machine Learning'. In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*. Vol. 108. Proceedings of Machine Learning Research, pp. 702–712.

Joseph, Matthew, Michael J. Kearns, Jamie H. Morgenstern and Aaron Roth (2016). 'Fairness in Learning: Classic and Contextual Bandits'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 325–333.

Kallus, Nathan and Angela Zhou (2018). 'Residual Unfairness in Fair Machine Learning from Prejudiced Data'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 2444–2453.

Kamiran, Faisal and Toon Calders (2009). 'Classifying without discriminating'. In: *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*. IEEE, pp. 1–6.

– (2012). 'Data preprocessing techniques for classification without discrimination'. In: *Knowledge and Information Systems* 33.1, pp. 1–33.

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

Kaplan, David (2008). *Structural equation modeling: Foundations and extensions*. Vol. 10. Sage Publications.

Kehrenberg, Thomas, Zexun Chen and Novi Quadrianto (2020a). 'Tuning Fairness by Balancing Target Labels'. In: *Frontiers in Artificial Intelligence* 3, p. 33. DOI: 10.3389/frai.2020.00033.

Kehrenberg, Thomas, Myles Bartlett, Oliver Thomas and Novi Quadrianto (2020b). 'Null-sampling for Interpretable and Fair Representations'. In: *European Conference on Computer Vision (ECCV)*. Glasgow, UK. DOI: 10.1007/978-3-030-58604-1.

Kehrenberg, Thomas, Viktoriia Sharmanska, Myles Bartlett and Novi Quadrianto (2021). 'Learning with Perfect Bags: Addressing Hidden Stratification with Zero Labeled Data'.

Khalifa, Muhammad, Hady Elsahar and Marc Dymetman (2021). 'A Distributional Approach to Controlled Text Generation'. In: *International Conference on Learning Representations (ICLR)*.

Kilbertus, Niki, Philip J. Ball, Matt J. Kusner, Adrian Weller and Ricardo Silva (2019). 'The Sensitivity of Counterfactual Fairness to Unmeasured Confounding'. In: *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*. Vol. 115. Proceedings of Machine Learning Research, pp. 616–626.

Kilbertus, Niki, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing and Bernhard Schölkopf (2017). 'Avoiding Discrimination through Causal Reasoning'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 656–666.

Kim, Byungju, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim and Junmo Kim (2019). 'Learning Not to Learn: Training Deep Neural Networks With Biased Data'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9012–9020. DOI: 10.1109/CVPR.2019.00922.

Kim, Hyunjik and Andriy Mnih (2018). 'Disentangling by Factorising'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 2654–2663.

Kingma, Diederik P. and Jimmy Ba (2015). 'Adam: A Method for Stochastic Optimization'. In: *International Conference on Learning Representations (ICLR)*.

Kingma, Diederik P. and Prafulla Dhariwal (2018). 'Glow: Generative Flow with Invertible 1x1 Convolutions'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10236–10245.

Kingma, Diederik P. and Max Welling (2014). 'Auto-Encoding Variational Bayes'. In: *International Conference on Learning Representations (ICLR)*.

Klambauer, Günter, Thomas Unterthiner, Andreas Mayr and Sepp Hochreiter (2017). 'Self-Normalizing Neural Networks'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 971–980.

Kleinberg, Jon, Sendhil Mullainathan and Manish Raghavan (2016). 'Inherent trade-offs in the fair determination of risk scores'. In: *arXiv preprint arXiv:1609.05807*.

Kohavi, Ron (1996). 'Scaling up the accuracy of Naive-Bayes classifiers: a Decision-Tree Hybrid'. In: *Knowledge Discovery and Data Mining*. Vol. 96, pp. 202–207.

Krauth, Karl, Edwin V. Bonilla, Kurt Cutajar and Maurizio Filippone (2017). 'AutoGP: Exploring the Capabilities and Limitations of Gaussian Process Models'. In: *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*.

Kusner, Matt J., Joshua R. Loftus, Chris Russell and Ricardo Silva (2017). 'Counterfactual Fairness'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 4066–4076.

Lampert, Christoph H., Hannes Nickisch and Stefan Harmeling (2009). 'Learning to detect unseen object classes by between-class attribute transfer'. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594.

Larochelle, Hugo, Dumitru Erhan and Yoshua Bengio (2008). 'Zero-data learning of new tasks'. In: *AAAI Conference on Artificial Intelligence*.

LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner (1998). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

LeCun, Yann, Corinna Cortes and Christopher J. C. Burges (1994). *The MNIST database of handwritten digits*.

Lee, Juho, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi and Yee Whye Teh (2019). 'Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 3744–3753.

Liu, Liyuan, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao and Jiawei Han (2020). 'On the Variance of the Adaptive Learning Rate and Beyond'. In: *International Conference on Learning Representations (ICLR)*.

Liu, Lydia T., Sarah Dean, Esther Rolf, Max Simchowitz and Moritz Hardt (2019). 'Delayed Impact of Fair Machine Learning'. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 6196–6200. DOI: 10.24963/ijcai.2019/862.

Liu, Ziwei, Ping Luo, Xiaogang Wang and Xiaoou Tang (2015). 'Deep Learning Face Attributes in the Wild'. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 3730–3738. DOI: 10.1109/ICCV.2015.425.

Locatello, Francesco, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf and Olivier Bachem (2019a). 'On the Fairness of Disentangled Representations'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 14584–14597.

Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf and Olivier Bachem (2019b). 'Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 4114–4124.

Locatello, Francesco, Ben Poole, Gunnar Raetsch, Bernhard Schölkopf, Olivier Bachem and Michael Tschannen (July 2020). 'Weakly-Supervised Disentanglement Without Compromises'. In: *International Conference on Machine Learning (ICML)*. Vol. 119. Proceedings of Machine Learning Research, pp. 6348–6359.

Lohaus, Michael, Michaël Perrot and Ulrike von Luxburg (2020). 'Too Relaxed to Be Fair'. In: *International Conference on Machine Learning (ICML)*. Vol. 119. Proceedings of Machine Learning Research, pp. 6360–6369.

Louizos, Christos, Kevin Swersky, Yujia Li, Max Welling and Richard S. Zemel (2016). 'The Variational Fair Autoencoder'. In: *International Conference on Learning Representations (ICLR)*.

Lum, Kristian and James Johndrow (2016). 'A statistical framework for fair predictive algorithms'. In: *arXiv preprint arXiv:1610.08077*.

Madras, David, Elliot Creager, Toniann Pitassi and Richard S. Zemel (2018). 'Learning Adversarially Fair and Transferable Representations'. In: *International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research, pp. 3381–3390.

Maity, Subha, Debarghya Mukherjee, Mikhail Yurochkin and Yuekai Sun (2020). 'There is no trade-off: enforcing fairness can improve accuracy'. In: *arXiv preprint arXiv:2011.03173*.

Mary, Jérémie, Clément Calauzènes and Noureddine El Karoui (2019). 'Fairness-Aware Learning for Continuous Attributes and Treatments'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 4382–4391.

Miyato, Takeru, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida (2018). 'Spectral Normalization for Generative Adversarial Networks'. In: *International Conference on Learning Representations (ICLR)*.

Nam, Jun Hyun, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee and Jinwoo Shin (2020). 'Learning from Failure: Training Debiased Classifier from Biased Classifier'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, pp. 20673–20684.

Oakden-Rayner, Luke, Jared Dunnmon, Gustavo Carneiro and Christopher Ré (2020). 'Hidden stratification causes clinically meaningful failures in machine learning for medical imaging'. In: *ACM CHIL '20: ACM Conference on Health, Inference, and Learning, Toronto, Ontario, Canada, April 2-4, 2020 [delayed]*, pp. 151–159.

Olteanu, Alexandra, Carlos Castillo, Fernando Diaz and Emre Kıcıman (2019). 'Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries'. In: *Frontiers in Big Data* 2, p. 13.

Oord, Aäron van den, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals and Alex Graves (2016). 'Conditional Image Generation with PixelCNN Decoders'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 4790–4798.

Pearl, Judea (2009). *Causality*. Cambridge university press.

– (2019). 'The seven tools of causal inference, with reflections on machine learning'. In: *Communications of the ACM* 62.3, pp. 54–60.

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pedreshi, Dino, Salvatore Ruggieri and Franco Turini (2008). 'Discrimination-aware data mining'. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 560–568.

Platt, John (1999). 'Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods'. In: *Advances in large margin classifiers* 10.3, pp. 61–74.

Quadrianto, Novi and Viktoriia Sharmanska (2017). 'Recycling Privileged Learning and Distribution Matching for Fairness'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 677–688.

Quadrianto, Novi, Viktoriia Sharmanska and Oliver Thomas (2019). 'Discovering Fair Representations in the Data Domain'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8227–8236. DOI: 10.1109/CVPR.2019.00842.

Radford, Alec, Karthik Narasimhan, Tim Salimans and Ilya Sutskever (2018). 'Improving language understanding by generative pre-training'.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever (2019). 'Language Models are Unsupervised Multitask Learners'.

Raghavan, Manish, Solon Barocas, Jon M. Kleinberg and Karen Levy (2020). 'Mitigating bias in algorithmic hiring: evaluating claims and practices'. In:

FAT '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pp. 469–481.

Rényi, Alfréd (1959). 'On measures of dependence'. In: *Acta Mathematica Academiae Scientiarum Hungarica* 10.3-4, pp. 441–451.

Rezende, Danilo Jimenez and Shakir Mohamed (2015). 'Variational Inference with Normalizing Flows'. In: *International Conference on Machine Learning (ICML)*. Vol. 37. JMLR Workshop and Conference Proceedings, pp. 1530–1538.

Roh, Yuji, Kangwook Lee, Steven Euijong Whang and Changho Suh (2021). 'FairBatch: Batch Selection for Model Fairness'. In: *International Conference on Learning Representations (ICLR)*.

Sagawa, Shiori, Pang Wei Koh, Tatsunori B Hashimoto and Percy Liang (2019). 'Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization'. In: *arXiv preprint arXiv:1911.08731*.

Sattigeri, Prasanna, Samuel C. Hoffman, Vijil Chenthamarakshan and Kush R. Varshney (2019). 'Fairness GAN: Generating datasets with fairness properties using a generative adversarial network'. In: *IBM Journal of Research and Development* 63.4/5, 3:1–3:9. DOI: 10.1147/JRD.2019.2945519.

Shu, Rui, Yining Chen, Abhishek Kumar, Stefano Ermon and Ben Poole (2020). 'Weakly Supervised Disentanglement with Guarantees'. In: *International Conference on Learning Representations (ICLR)*.

Sohoni, Nimit Sharad, Jared Dunnmon, Geoffrey Angus, Albert Gu and Christopher Ré (2020). 'No Subclass Left Behind: Fine-Grained Robustness in Coarse-Grained Classification Problems'. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Sriperumbudur, Bharath K. and Gert R. G. Lanckriet (2009). 'On the Convergence of the Concave-Convex Procedure'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1759–1767.

Stiennon, Nisan et al. (2020). 'Learning to summarize with human feedback'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, pp. 3008–3021.

Suter, Raphael, Đorđe Miladinovic, Bernhard Schölkopf and Stefan Bauer (2019). 'Robustly Disentangled Causal Mechanisms: Validating Deep Representations for Interventional Robustness'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 6056–6065.

Tarleton, Nick (2010). 'Coherent extrapolated volition: A meta-level approach to machine ethics'. In: *Machine Intelligence Research Institute*.

Thanh, Binh Luong, Salvatore Ruggieri and Franco Turini (2011). 'k-NN as an implementation of situation testing for discrimination discovery and prevention'. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pp. 502–510. DOI: 10.1145/2020408.2020488.

Tolan, Songül (2019). 'Fair and Unbiased Algorithmic Decision Making: Current State and Future Challenges'. In: *arXiv preprint arXiv:1901.04730*.

Tsybakov, Alexander B et al. (2004). 'Optimal aggregation of classifiers in statistical learning'. In: *The Annals of Statistics* 32.1, pp. 135–166.

Ustun, Berk, Yang Liu and David C. Parkes (2019). 'Fairness without Harm: Decoupled Classifiers with Preference Guarantees'. In: *International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research, pp. 6373–6382.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin (2017). 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008.

Verma, Sahil and Julia Rubin (2018). 'Fairness definitions explained'. In: *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, pp. 1–7.

Wick, Michael L., Swetasudha Panda and Jean-Baptiste Tristan (2019). 'Unlocking Fairness: a Trade-off Revisited'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8780–8789.

Wightman, Linda F (1998). 'LSAC National Longitudinal Bar Passage Study'. In: LSAC Research Report Series.

Woodworth, Blake E., Suriya Gunasekar, Mesrob I. Ohannessian and Nathan Srebro (2017). 'Learning Non-Discriminatory Predictors'. In: *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*. Vol. 65. Proceedings of Machine Learning Research, pp. 1920–1953.

Wu, Yongkai, Lu Zhang, Xintao Wu and Hanghang Tong (2019). 'PC-Fairness: A Unified Framework for Measuring Causality-based Fairness'. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3399–3409.

Xian, Yongqin, Christoph H Lampert, Bernt Schiele and Zeynep Akata (2018). 'Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.9, pp. 2251–2265.

Xiao, Han, Kashif Rasul and Roland Vollgraf (2017). 'Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms'. In: *arXiv preprint arXiv:1708.07747*.

Xiao, Taihong, Jiapeng Hong and Jinwen Ma (2018). 'DNA-GAN: Learning disentangled representations from multi-attribute images'. In: *ICLR workshop*.

Yudkowsky, Eliezer (2011). 'Complex Value Systems in Friendly AI'. In: *Artificial General Intelligence*, pp. 388–393.

Zafar, Muhammad Bilal, Isabel Valera, Manuel Gomez-Rodriguez and Krishna P. Gummadi (2017a). 'Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment'. In: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pp. 1171–1180. DOI: 10.1145/3038912.3052660.

– (2017b). 'Fairness Constraints: Mechanisms for Fair Classification'. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. Vol. 54. Proceedings of Machine Learning Research, pp. 962–970.

Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov and Alexander J. Smola (2017). 'Deep Sets'. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3391–3401.

Zemel, Richard S., Yu Wu, Kevin Swersky, Toniann Pitassi and Cynthia Dwork (2013). 'Learning Fair Representations'. In: *International Conference on Machine Learning (ICML)*. Vol. 28. JMLR Workshop and Conference Proceedings, pp. 325–333.

Zhang, Brian Hu, Blake Lemoine and Margaret Mitchell (2018). 'Mitigating Unwanted Biases with Adversarial Learning'. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '18. New Orleans, LA, USA, pp. 335–340. ISBN: 9781450360128. DOI: 10.1145/3278721.3278779.

Zhang, Quanshi and Song-Chun Zhu (2018). 'Visual interpretability for Deep Learning: a survey'. In: *Frontiers of Information Technology & Electronic Engineering* 19.1, pp. 27–39.