LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

mit media lab

MASTER THESIS

Towards Fair Budget-Constrained Machine Learning

A thesis submitted for the degree
Master of Science

Duy Patrick Tu

Reviewer: Prof. Dr. Andreas Butz
Ludwig-Maximilians-Universität München (LMU)

Advisor: Michiel Bakker, M. Sc.
Massachusetts Institute of Technology (MIT)

Date: 14. April 2020

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. April 2020

. . . .

(Unterschrift des Kandidaten)

Abstract

Machine learning systems are increasingly deployed in high-stake situations such as criminal justice, credit risk assessment, and medical diagnoses. With predictive decisions significantly impacting crucial aspects of individuals' life, society raised concerns about unfair treatment and discrimination by such algorithmic tools. Consequently, a growing body of research has established around fairness in machine learning. While the literature focused mainly on a setting, in which all features are ready at hand, this work focuses on a setting known as prediction-time active feature-value acquisition (AFA). Here, a decision maker can sequentially query information (features) at some cost and further makes a final prediction upon it. The aim of this cumulative thesis is to investigate algorithmic fairness in prediction-time AFA settings.

The contributions of this work are twofold. First, a framework for choosing a set of confidence-based stopping criteria is proposed to redistribute information (feature) budgets among individuals. Often, individuals from underrepresented groups in the data will face a higher likelihood of erroneous decisions. Naturally, this framework encourages collecting more information for these individuals to ensure equally confident decisions. Using a calibrated probabilistic classifier, our experiments demonstrated single error parity (equal opportunity) in addition to calibration by groups.

Second, staying in the AFA domain, we translate the problem into a Markov decision process and train a reinforcement learning agent to sequentially choose subsets of features that are predictive for the outcome but do not increase the demographic disparity. This is done by incorporating an adversary in the reward function, that penalizes the agent if it selects *unfair* features. By tuning a hyperparameter representing the magnitude of fairness, the framework is able to trade off predictive performance and fairness (demographic parity), which we confirmed experimentally.

Acknowledgements

First and foremost, I would like to thank my advisor Michiel who onboarded me on this stint in research at the MIT Media Lab and who had the trust, confidence and patience to work with me. I still cannot believe how much I have learned and grown throughout this journey. I would like to thank him for the constant support and guidance.

Further, I would like to thank my mentor and reviewer Prof. Dr. Andreas Butz for the support of conducting my Master thesis in joint collaboration with the MIT.

I would also like to thank Prof. Alex "Sandy" Pentland for receiving me at the Human Dynamics Group at the MIT Media Lab, and all the people who made my stay at MIT a pleasant experience - Humberto, Daoud, Alejandro, Luis, Dalia, Gianni, Vikki, Rubez, Paul and Katherina.

In particular, I would like to thank my parents Minh and Tuyet for always supporting me unconditionally in every situation and providing me financial stability during my studies. Also, I want to thank my sister, my friends and especially Alyssa for all the emotional support and encouragement throughout that time and beyond. Moreover, I want to thank all the other collaborators, who made this research fruitful and who I owe the participation in two great academic conferences.

Finally, I would like to thank the German Academic Exchange Service (DAAD) and the Bavarian State for generously supporting me with the PROSA Scholarship during my time in Cambridge.

Contents

Chapter 1

Introduction

1.1 Motivation

The recent decade has seen a rise in artificial intelligence, which was predominantly powered by machine learning-based systems. The progress made at the frontier of machine learning research is nothing but remarkable. We have seen breakthroughs in intelligent systems that surpass human-level performance at a variety of tasks such as in recognizing images [1], playing the game of Go [2] or identifying breast cancer based on mammograms [3]. As the "new electricity", these transformative technologies are being widely adopted in various industries with the promise of creating economic value [4]. Hence, machine learning models are becoming more and more integrated into our daily lives.

This may range from low-stake situations such as setting up an alarm using a conversational assistant to high-stake situations where an algorithm predicts the creditworthiness of an individual in a loan application. As you can imagine, the second scenario has a significantly higher impact on an individual's life. Wrong decisions may have drastic and irreversible consequences in the life of that individual, such as existential bankruptcy. Other high-stake applications of machine learning can be observed in criminal risk assessment, job screening, welfare fraud detection and medical diagnosis.

While there is no doubt about the utility and efficiency of automated systems, society at large has raised concerns about discrimination and unfairness within those systems. What if a model's output results in decisions that are systematically biased against minority groups? What if historical or real-world biases in the data are perpetuated by those models? This would lead to unfairly denied loans, missed employment opportunities or even jail sentences. Indeed, there is substantial evidence of discrimination by deployed machine learning systems, which has been covered in the mainstream media. A criminal recidivism risk scoring tool used in courtrooms called *COMPAS* has been claimed to encode racial bias. A study by ProPublica found that African-American defendants were mistakenly labeled as potential recidivists at a higher rate than white Americans [5]. Reuters reported in 2018, that a machine learning-based recruiting tool for CV screening was discontinued by Amazon as it "showed bias against women" [6]. The *Gender Shades* study showed how commercial facial recognition software by IBM, Google, Microsoft significantly "misgender" women of color [7]. Facebook was charged in March 2019 by the US Department of Housing and Urban Development (HUD)[1] for violating the Fair Housing Act as targeted ads discriminated based on protected attributes like race, sex, and nationality [8]. In February 2020, a Dutch risk scoring system called *SyRI*, which

[1] `https://www.hud.gov/sites/dfiles/Main/documents/HUD_v_Facebook.pdf`

aims at predicting the likelihood of social security claimants to commit benefits or tax fraud, was sentenced for breaching human rights as it discriminated against low-income neighborhoods [9].

The potential harm of these data-driven methods led to an exploding interest by the academic community to study fairness in machine learning. Fairness, accountability and transparency became key ingredients towards more *human-centric* machine learning. Substantial efforts have been made in order to define and quantify fairness within this context, to understand the causes of discrimination and to develop techniques to mitigate unfairness in algorithmic systems. These efforts are also in line with the General Data Protection Regulation (GDPR), which was put in place by the European Union in May 2019. In particular, the GDPR demands privacy-preserving, fair and transparent processing of personal data[2].

Most methods that are aimed at achieving *fairer* predictive models focus on the standard classification setting, in which data is assumed to be complete and readily available. While this assumption might hold in course projects and data science competitions, the real world often deals with missing values, incomplete data sets, and situations where information can be specifically collected at some additional cost. Consider a courtroom setting, where a defendant is accused of having committed a crime. Initially, there might be insufficient information about the case in order for a judge or jury to make an informed decision. Hence, in a pre-trial investigation, facts are collected by e.g. police officers or public procurators. Sequentially, more and more information is queried at some administrative cost by inviting witnesses, collecting evidences and conducting cross-examinations. This can carry on across several trial stages until sufficient evidence is collected to make a confident prediction, the judgment sentence. In machine learning, this setting is known as *prediction-time active feature-value acquisition* (AFA) [10] and is relevant in many high-stake domains.

To the best of our knowledge, there is only one paper [11] exploring the intersection of fairness in machine learning and AFA systems. In their work, a decision maker can sequentially acquire features at prediction-time adapted to subgroup-level needs in order to balance disparities in false positive or false negative rates across subgroups. In order to determine the subgroup-specific information budgets, they utilize an optimization framework. The goal of this thesis is to extend this line of work and develop further methods to promote fairness in budget-constrained, sequential decision making. This work is different from [11] in that it does not rely on optimization methods to determine information budgets but rather suggests two dynamic approaches for determining final feature sets using (1) confidence-based stopping criteria and (2) an adversarial reinforcement learning framework.

1.2 Key Contributions

This thesis presents two novel methods to mitigate unfairness in budget-constrained machine learning problems. Thus, the key contributions are twofold:

[2] `https://gdpr-info.eu/art-5-gdpr/`

- First, we propose a framework for choosing a set of stopping criteria based on the probabilistic confidence of a classifier that leads to fairness in terms of single error parity (equal opportunity) in addition to calibration by groups. The framework scales efficiently to possibly multiple intersections of sensitive attributes as long as we have access to the base rate of those subgroups.
- Second, we propose a unified framework where the AFA setting is modeled as a Markov decision process and a reinforcement learning agent sequentially selects a subset of *fair* features. This is done by involving an adversary that penalizes the agent to acquire features that lead to demographic disparity. By tuning a hyperparameter representing the magnitude of fairness in the reward function, the framework is able to effectively trade off predictive performance and demographic parity.

1.3 Thesis Structure

The structure of this thesis is organized as follows. **Chapter 1** introduces the problem of interest and motivates the setting. Further, the main contributions are outlined.

Chapter 2 introduces some relevant background knowledge and provides an overview of machine learning concepts, i.e. supervised and reinforcement learning. It further introduces the AFA setting and discusses some related work on fairness in machine learning.

Chapter 3, titled "On Fairness on Budget-Constrained Decision Making" investigates how individual information (features) budgets across protected groups can provide fairness guarantees. The fairness measure of interest here is calibration and equal opportunity. In particular, the overarching question is when to stop acquiring additional features to provide fair and accurate decisions. The chapter draws on a workshop paper presented at the *KDD 2019 Workshop on Explainable AI (XAI) for Fairness, Accountability and Transparency* (Bakker, Noriega-Campero, Tu et al. 2019). A further expanded version of this work is currently under review at the *International Conference on Machine Learning (ICML) 2020.*

Building on the insights of the previous chapter, **Chapter 4** titled "DADI: Dynamic Discovery of Fair Information with Adversarial Reinforcement Learning" aims to combine two aspects of the AFA pipeline, feature acquisition strategy and stopping criterion, to ensure fairness in terms of demographic parity for classification tasks downstream. This work was accepted at the *NeurIPS 2019 Human-Centric Machine Learning Workshop* (Bakker, Tu et al. 2019) and a further extended version is currently under review at the *International Joint Conference on Artificial Intelligence (IJCAI) 2020.*

Finally, **Chapter 5** concludes this thesis with a discussion and motivates future work.

Chapter 2

Background and Related Work

This chapter introduces some general background knowledge on machine learning and discusses some related work. This lays the groundwork of the research contributions in chapter 3 and 4. First, the concepts of supervised and reinforcement learning are introduced. Second, we lay out the setting of prediction-time active feature-value acquisition. Last, we elaborate on fairness in machine learning and approaches to achieve statistical notions of fairness.

2.1 Supervised Learning

Supervised learning has been driving most of the applications in machine learning. The idea of supervised learning is to learn an unknown functional relationship between inputs and outputs. The *supervised* terminology arose as the process of an algorithm learning from a training data set resembles a teacher supervising the learning process. Knowing the correct answers, the so-called *labels* (or sometimes called *targets* or *outcomes*), the algorithm iteratively outputs predictions on the training data and is "corrected" by the teacher in order to improve itself.

2.1.1 Empirical Risk Minimization

Following the notation in [12, 13], let $\mathcal{X}$ be an input domain of individuals and let $\mathcal{Y}$ be a target domain. The instances in the input domain $\mathcal{X}$, which are represented by an n-dimensional feature vector $\mathbf{x} = (x_1, ..., x_n)$, and the label y from the target space $\mathcal{Y}$ are assumed to be drawn from some unknown probability distribution $\mathcal{P}$ over the joint space $\mathcal{X} \times \mathcal{Y}$. Given a sequence of labeled training examples $\mathcal{D}_{train} = ((\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(m)}, y^{(m)}))$, where each example $(\mathbf{x}^{(i)}, y^{(i)})$ of individual i is drawn i.i.d from $\mathcal{P}$ and denoted as a pair of a feature vector $\mathbf{x}^{(i)}$ and its corresponding label $y^{(i)}$, the goal is to learn a mapping $h : \mathcal{X} \mapsto \mathcal{Y}$ that best approximates the unknown functional relationship $y = f(\mathbf{x})$. This is true for a predictor h that minimizes the so-called true risk $\mathcal{R}_{true}(h) = P_{(\mathbf{x},y)\sim\mathcal{P}}[h(\mathbf{x}) \neq f(\mathbf{x})]$, the probability of predicting the wrong label on a randomly drawn sample $\mathbf{x}$ from the distribution $\mathcal{P}$. However, since the true distribution $\mathcal{P}$ is usually unknown, the true risk is not available to a potential learning algorithm. Instead, in practice, the empirical risk $\mathcal{R}_{emp}$ (or sometimes called training error) is used as a proxy for the true risk. A popular corresponding learning paradigm is empirical risk minimization (ERM), where based on the available training data $\mathcal{D}_{train}$ the empirical risk is minimized to find an optimal predictor h^*, i.e.,

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_{emp}(h) = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{|\mathcal{D}_{train}|} \sum_{(\mathbf{x},y) \in \mathcal{D}_{train}} l(h(\mathbf{x}), y). \tag{2.1}$$

The hypothesis space $\mathcal{H}$ is the set of all possible candidate predictors, and l denotes a suitable loss function that measures the discrepancy between the predicted value from a candidate model h and the actual ground truth label y of a given instance. ERM seeks to find an optimal predictor h^* that achieves the minimal average loss over all instances of the training set $\mathcal{D}_{train}$. Although this seems to be a natural way to achieve good results, this approach also carries pitfalls. In theory, ERM can return a predictor $h(\mathbf{x}) = y, \forall(\mathbf{x}, y) \in \mathcal{D}_{train}$ that simply memorizes the labels for each observation in the training set. While this solution results in an optimal empirical risk $\mathcal{R}_{emp}(h) = 0$, the model would not necessary perform well on unseen observations $(\mathbf{x}, y) \notin \mathcal{D}_{train}$ in the real world. In general, the desired goal is that the model is able to generalize beyond the training set, minimizes the true risk and approximates the true underlying relationship $y = f(\mathbf{x})$.

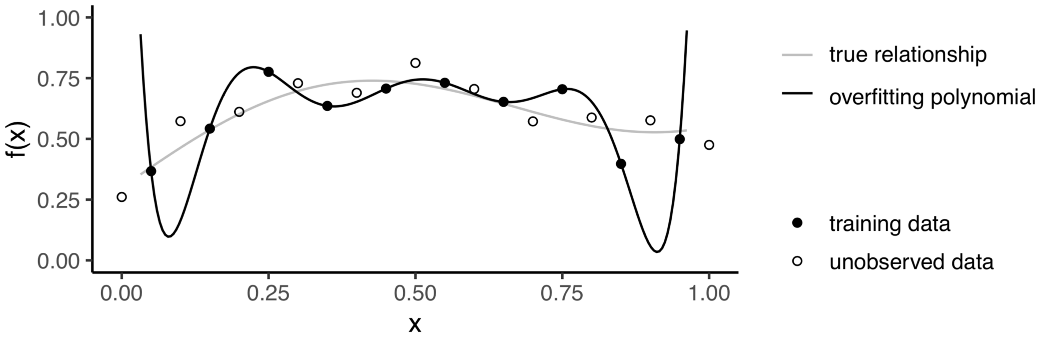


Figure 2.1: *Overfitting:* The black line represents a polynomial function that is able to fit every single training data point. However, this does not represent a good approximation of true distribution as the residuals on the unobserved data points are high [12].

Consider the example illustrated in Figure 2.1, where the data follows a polynomial distribution with some random noise. Suppose $\mathcal{H}$ represents the class of all polynomial functions. From $\mathcal{H}$ it is possible to find a polynomial function with a sufficiently high degree M such that it fits each training data point and hence achieves zero empirical risk. However, the error on data points from the distribution which the learning algorithm has not seen yet is considerably higher. This phenomenon is called *overfitting.* To reduce overfitting, several methods can be applied such as restricting the complexity of the hypothesis class $\mathcal{H}$ or introducing a regularization term $J(h)$ in the objective function that penalizes complex models where λ is a hyperparameter controlling the magnitude of the penalty.

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_{emp}(h) + \lambda * J(h) \tag{2.2}$$

2.1.2 Types of Supervised Learning

Supervised learning can be divided into two general tasks: *classification* and *regression*. The fundamental difference between these two techniques is the property of the target space $\mathcal{Y}$. Regression denotes the problem of learning a continuous output with $\mathcal{Y} = \mathbb{R}$, e.g. predicting the temperature at a point in time. In contrast, if the target space $\mathcal{Y}$ is discrete and the goal is to assign based on the input one of n labels (e.g. sunny, cloudy or rainy), the task is called classification. Here, we further distinguish between *multi-class classification* when $|\mathcal{Y}| > 2$ and *binary classification*, e.g. $\mathcal{Y} = \{0, 1\}$. In line with the vast majority of work exploring algorithmic fairness in supervised learning problems, we mostly focus on binary classification in this work. Following the categorization in [12], there are three prediction types within binary classification:

- *Discrete classifiers* predict the class membership of an observation and directly assign the label to a given observation.
- *Scoring classifiers* predict a real-valued risk score. Concrete label predictions can be obtained by thresholding. However, in most cases raw real-valued scores are rather difficult to interpret.
- *Probabilistic classifiers* try to estimate a probability distribution over all classes given the data point. Probabilities can be obtained by calibrating risk scores. Also in this case thresholding can be applied to determine the class prediction.

2.1.3 Performance Metrics

After learning a model based on some observed data, the next step is to find out how well a model performs. In order to quantify success, several performance metrics have been proposed to evaluate machine learning models. One widely adopted approach is to apply the performance metric to a hold-out test data set, which the algorithm has not been exposed to during the learning process. In the following, common performance measures for the different supervised tasks are introduced.

Regression In regression, performance is usually quantified using residual-based approaches. Let $h(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ be a fitted regression model. The residual $\epsilon = y - h(\mathbf{x})$ is the difference between target and predicted value of the model. One commonly used measure is the mean squared error (MSE). It is the average of the squared residuals across all considered observations.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h(\mathbf{x}^{(i)}))^2 = \frac{1}{m}\sum_{i=1}^{m}(\epsilon^{(i)})^2 \tag{2.3}$$

As the residuals are squared in the MSE, large discrepancies due to outliers have a big effect on our performance metric. A more robust alternative towards outliers is the mean absolute error (MAE),

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y^{(i)} - h(\mathbf{x}^{(i)})| = \frac{1}{m}\sum_{i=1}^{m}|\epsilon^{(i)}|. \tag{2.4}$$

Discrete Classifiers Discrete classifiers in the context of binary classification assign an instance to one of the two binary outcomes. The two outcome labels are also often referred to as the negative and positive class. In general, the aim is to be able to discriminate well between those two classes and predict the correct class label. From the so-called *confusion matrix* shown in Table 2.1, a multitude of performance measures for discrete classifiers can be derived. For a binary classifier $h(\mathbf{x}) : \mathcal{X} \mapsto \{0, 1\}$, the confusion matrix summarizes possible outcomes of predicted labels and ground truth labels. In the following, an overview of commonly used performance measures is given. We refer to [14] for a more exhaustive overview.

		Ground Truth Label	
		$y = 1$	$y = 0$
Predicted	$h(\mathbf{x}) = 1$	True Positive (TP)	False Positive (FP)
Label	$h(\mathbf{x}) = 0$	False Negative (FN)	True Negative (TN)
	Total	TP + FN	FP + TN

Table 2.1: 2 × 2 confusion matrix

The most well known and widely used evaluation metric is the accuracy rate (*Acc*). The accuracy measures the percentage of correct predictions of the classifier on the data.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.5}$$

However, not in all applications being correct on the positive versus the negative class is equally important. The recall (*Rec*), also called true positive rate (TPR), measures the percentage of samples from only the positive class that were correctly predicted as positive.

$$Rec = TPR = \frac{TP}{TP + FN} \tag{2.6}$$

Analogously, the specificity (*Spe*) measures the share of correctly predicted negative examples and is hence also called true negative rate (TNR).

$$Spe = TNR = \frac{TN}{TN + FP} \tag{2.7}$$

The precision (*Pre*), also called positive predictive value (PPV), is an estimate of the probability that a positively predicted instance is correct. It is the proportion of true positives across all positively predicted instances.

$$Pre = PPV = \frac{TP}{TP + FP} \tag{2.8}$$

The metrics introduced above, measure mainly the correctly classified portion of the classifier. Naturally, they also have complements measuring the incorrect portion. Their corresponding complements are the error rate ($Err = 1 - Acc$), the false positive rate ($FPR = 1 - TNR$), the false negative rate ($FNR = 1 - TPR$) and the false discovery rate ($FDR = 1 - PPV$).

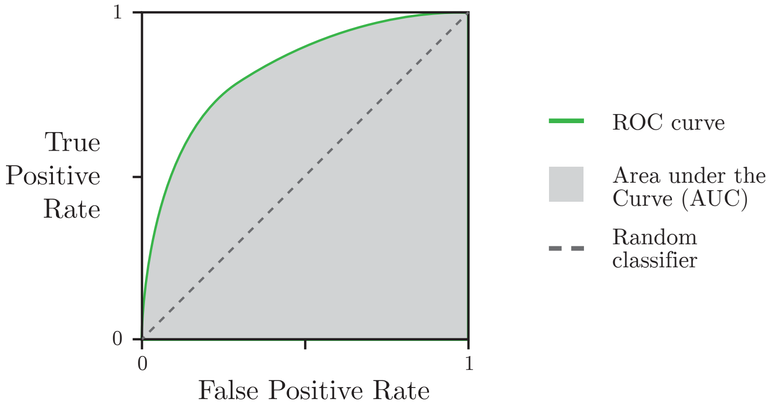


Figure 2.2: *ROC curve:* Each point on the green solid curve is a scoring classifier applied with a different threshold t. The dashed black line is the random baseline classifier.

Scoring Classifiers Let $s(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ be a model which predicts a risk score that can possibly range from $-\infty$ to ∞. As scores from different classifiers might be scaled differently, it is important to note that the order of the scores is more relevant than the actual value when comparing different scoring classifiers. In practice, one can set a threshold value t to assign the final discrete class labels, i.e.,

$$h(\mathbf{x}) = \begin{cases} 0 & s(\mathbf{x}) < t \\ 1 & else. \end{cases} \tag{2.9}$$

Converting scoring classifiers into discrete classifiers by thresholding enables us to apply the evaluation measures derived from the confusion matrix introduced in the previous paragraph. However, different thresholds will lead to different confusion matrices. The receiver operating characteristics (ROC) curve is a way to take this into consideration. The ROC curve visualizes the pairs TPR and FPR for each possible threshold t. A natural way to measure the performance of this classifier is to compute the area under the ROC curve (AUC). A perfect classifier would be able to achieve $AUC = 1$ while a completely random guessing classifier would have $AUC = 0.5$. The ROC curve shows that the choice of the threshold t trades off the TPR with FPR.

Probabilistic Classifiers Let $\pi(\mathbf{x}) : \mathcal{X} \mapsto [0, 1]$ be a probabilistic classifier. Similarly to scoring classifiers, probabilistic classifiers also output a real-valued number, however in the interval between 0 and 1. In fact, scores can be transformed into probabilities using scaling methods such as Platt scaling [15] or isotonic regression [16]. Hence, the same evaluation measures introduced in the previous paragraph such as thresholding and the AUC can be applied.

However, there exists a further desirable property besides discriminatory power for probabilistic classifiers: *calibration*. Well-calibrated probability predictions of an event reflect the true probability as close as possible. Formally, a classifier is perfectly calibrated when $P(y = 1|\pi(\mathbf{x}) = p) = p$. Intuitively this means that if there are 100 instances where the model predicts with probability $p = 0.6$ the positive class, 60 of them will actually belong to the positive class.

One proposed measure that takes calibration into account is the Brier score (BS) [17]:

$$BS = \frac{1}{m} \sum_{i=1}^{m} (y^{(i)} - \pi(\mathbf{x}^{(i)}))^2. \tag{2.10}$$

The BS is the average mean squared difference of predicted probabilities π and actual outcomes y and can be thought of as an *accuracy* measure of the probabilistic predictions. The best achievable BS is 0 while the worst achievable is 1.

2.1.4 Artificial Neural Networks

Artificial neural networks are popular and powerful function approximators for supervised learning, which date back to the early work of [18, 19]. They were initially inspired by their biological counterparts in the brain. Biological neural networks comprise billions of neurons that are interconnected and process electrochemical signals. The simplified, high-level idea is that every single neuron receives several input signals from preceding neurons and *fires* through its single output when a certain *action potential* is reached during information processing. The output signal is then forwarded to the connected neurons downstream [20].

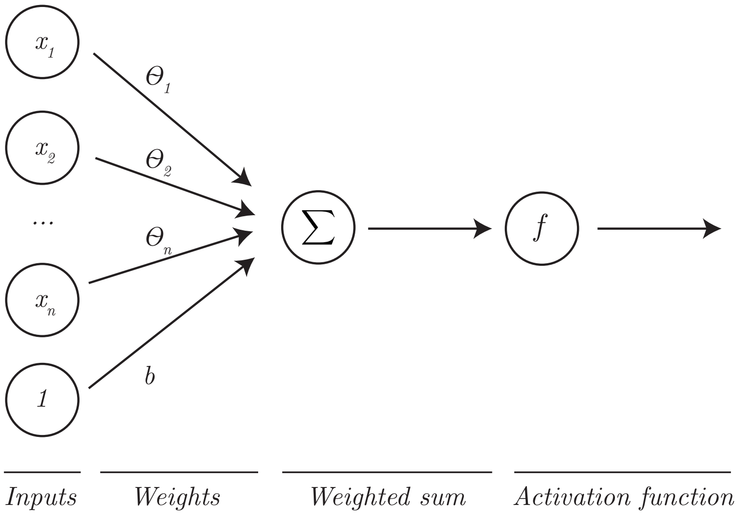


Figure 2.3: *Perceptron:* The perceptron takes input data and computes the weighted sum. The sum is further processed through an activation function f.

Artificial neural networks formalize this core idea mathematically. We follow the notation in [21] to introduce artificial neural networks. If clear from the context, we omit the prefix *artificial* and refer solely to neural networks. The basic unit of the neural network is the artificial neuron, also called *perceptron*. A perceptron has n input links. Further, each link also has some numeric weight θ_i associated with it, which represents the strength of each connection. The inputs are processed by computing the weighted sum and a bias b. Subsequently, an activation function f is applied to the scalar value to derive the output of the neuron:

$$f(\sum_i^n \theta_i x_i + b). \tag{2.11}$$

Typically, non-linear activation functions are utilized to enable the approximation of non-linear functions. A common activation function is the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{2.12}$$

A further widely used activation function is the rectified linear unit (ReLU) [22]:

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0. \end{cases} \tag{2.13}$$

Neural networks can be designed by composing multiple perceptron units in a chain. In a *feed-forward network*, connections only face forward in one direction forming a directed, acyclic graph. Feed-forward networks are usually organized in layers. Each neuron of a layer receives its inputs from neurons of the preceding layer. Between the input layer that processes the data input and the output layer, further so-called *hidden layers* of neurons can be introduced. The length of the chained layers corresponds to the depth of the model, which led to the terminology *deep learning* and *deep neural networks.*

The goal of the learning process of neural networks is to find a set of weights θ that approximates the true distribution and minimizes the loss function. Based on the ground truth labels of the training data, the weight parameters can be updated iteratively. The learning process can be illustrated in the following steps:

1. **Forward pass:** An instance from the data set is inputted to the network and forwarded across the nodes to get a predicted output $\hat{y} = f(\mathbf{x}; \theta)$ for each sample.
2. **Loss evaluation:** The quality of the prediction $\hat{y} \in \mathcal{Y}$ is assessed by comparing it to their ground truth label. This is done by using a suitable loss function e.g.
   - **Logistic loss:** $L(\theta) = \sum_{i=1}^{m} \log(1 + \exp(-y \cdot \hat{y}))$
   - **L2 loss:** $L(\theta) = \frac{1}{2} \sum_{i=1}^{m} (y - \hat{y})^2$.
3. **Backpropagation of gradients:** In order to minimize the respective loss function, the gradients of the loss function with respect to the weights are computed. The backpropagation algortihm [24] does this by applying the chain rule. The chain rule computes derivatives of compositions of multiple functions. Given the functions $y = g(x)$ and $z = f(g(x))$, the derivative $\frac{\delta z}{\delta x}$ can be obtained according to

   $$\frac{\delta z}{\delta x} = \frac{\delta z}{\delta y} \cdot \frac{\delta y}{\delta x}. \tag{2.14}$$

   Using the chain rule, the local gradients with respect to each weight θ_i can be backpropagated through the whole network. Notably, in contrast to the forward pass, this is a backwards pass in the opposite direction until the earliest layer is reached.
4. **Weight update:** A widely used method to update the weights with respect to each

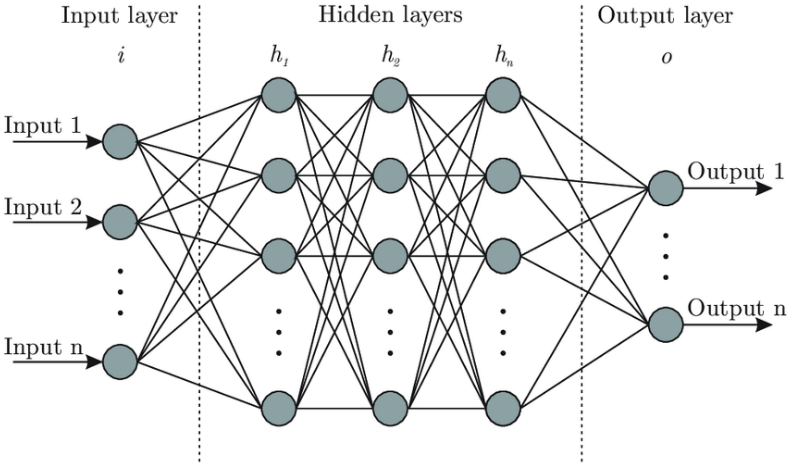


Figure 2.4: *Feed-forward neural network:* A fully connected neural network with 3 hidden layers, where each neuron of a layer is connected to each neuron of the preceeding layer [23].

network parameter is *stochastic gradient descent*, where the gradients are updated just based on a subset of the training data. Gradient descent updates the weights in the negative direction of the gradient loss, i.e.,

$$\theta_i \leftarrow \theta_i - \alpha \frac{\delta L(\theta)}{\delta \theta_i} \tag{2.15}$$

where α represents the learning rate and determines how big the steps in the direction of the local minimum are.

2.1.5 Decision Trees and Random Forest

Another popular algorithm in supervised learning is random forest [25]. The random forest belongs to the family of ensemble methods, which combine the predictions of multiple base models to achieve a more accurate prediction than a single, isolated model. In particular, it is a bootstrap aggregation or *bagging* method, which aggregates the output of several models fitted to different bootstrap samples of the training data. This reduces the variance in the predictions and hence can mitigate overfitting. As random forests utilize decision trees as base models, we first introduce classification and regression trees (CART) [26].

Classification and Regression Trees Decision trees are based on the idea to conclude a prediction based on a sequence of simple binary decisions that are organized in a hierarchical, tree-like structure. More formally, following the notation in [27, 28], a decision tree T is a directed, acyclic graph that consists of a set $\mathcal{N}$ of nodes and a set $\mathcal{E}$ of edges. Each node corresponds to a (binary) decision in the feature space and is connected via edges to a maximum of one parent node and a minimum of two child nodes. The node on top of the tree is called the root node while nodes at the bottom are referred to as leaf nodes. A data point traverses the tree from the root node to a leaf following a unique path which is determined by each decision of the traversing nodes. In particular, a decision node splits incoming samples according to a splitting function in two disjoint subsets, which corresponds to sending the samples to a subsequent child node. A common choice is to split along one input dimension j of the input space $\mathcal{X}$ using a threshold t, e.g. $N_{child1} = \{(\mathbf{x}, y) \in N_{parent} : x_j \geq t\}$ and $N_{child2} = \{(\mathbf{x}, y) \in N_{parent} : x_j < t\}$, where x_j corresponds to the value of the j-th feature of a corresponding data point. After the learning phase, the training instances that reached a particular leaf node are used to model a posterior distribution. This posterior distribution of a leaf node is used to make a prediction at inference time for incoming samples that end up in that particular leaf node.

The learning process of a tree focuses on finding the *optimal structure* of the tree, i.e. which feature dimension j to chose for splitting at each node and the value of the threshold t for the split. There are different impurity measures that define the optimality criterion based on the nature of a task. Suppose that there are $|T|$ leaf nodes indexed by $\tau = 1, ..., |T|$. In regression tasks with continuous targets the aim is to find a structure that minimizes the sum of squared errors across leaf nodes, i.e.,

$$Q_\tau(T) = \sum_{(\mathbf{x},y) \in N_\tau} (y - \bar{y}_{N_\tau})^2 \tag{2.16}$$

where $\bar{y}_{N_\tau} = \frac{1}{|N_\tau|} \sum_{(\mathbf{x},y) \in N_\tau} y$ represents the arithmetic mean of the continuous targets of the data points that fall in the region defined by leaf node N_τ. For classification problems with K targets, different measures exist to assess the impurity a node N. Two commonly used choices are the Gini Index

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k}(1 - p_{\tau k}) \tag{2.17}$$

and the cross entropy

$$Q_\tau(T) = -\sum_{k=1}^{K} p_{\tau k} \ln p_{\tau k} \tag{2.18}$$

where $p_{\tau k}$ is the proportion of data points that belong to class k in the region defined by the node N_τ. Both measures encourage regions that have a high proportion of data points belonging to only one class.

Generally, the optimization problem of finding the optimal structure of a tree with respect to an impurity measure is computationally infeasible due to the combinatorially large solution space. Instead, a greedy optimization is usually applied in practice, where, starting with a single root node representing the whole input space, further nodes are added sequentially. At each step, a pair of leaf nodes are added from a number of candidate regions to the existing tree which corresponds to choosing which of the feature dimension to split and specifying the threshold value. This joint optimization over the choice of input variable and threshold can be efficiently solved using exhaustive search. The optimal split is the one that minimizes the impurity measure of the resulting leaf nodes.

Iteratively, the tree can be grown using this greedy strategy until a stopping criterion is met. There are three commonly used stopping criteria: (1) maximal tree depth, (2) minimum population per leaf, and (3) minimum variation of the impurity objective.

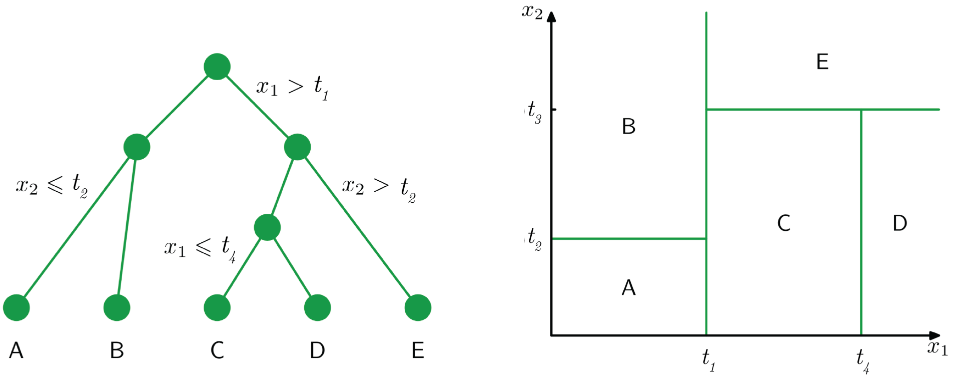


Figure 2.5: *Left*: Each node of the decision tree splits a data point $\mathbf{x}$ along a input feature dimension x_i using threshold t_i. *Right:* Each leaf node represents a region in the feature space that is partitioned by the thresholds [27].

The first criterion naively stops the iterative splitting process when the depth of the tree reaches a maximum depth. The second criterion concerns the number of training examples that end up in a leaf. If the population per leaf node is below a certain threshold, the splitting terminates. The last criterion considers the impurity measure that is optimized. If the additional improvement in impurity is below a certain threshold, further splitting is stopped due to the limited information gain.

Eventually, the leaves of a tree and i.e. the training examples that reach that particular node are used to model the posterior distribution. Predictions on new unseen examples can be performed by passing them downwards the tree until they reach a leaf node N_τ. The posterior model stored in the leaf node N_τ can be used for the prediction e.g. in the case of regression by taking the average target values of the observations of the leaf

$$\hat{y} = \frac{1}{|N_\tau|} \sum_{(\mathbf{x},y)\in N_\tau} y \tag{2.19}$$

or in the case of classification by choosing the class that has the largest proportion in that node

$$\hat{y} = \arg\max_k p_{\tau k}. \tag{2.20}$$

Random Forest While single decision trees have favorable properties such as interpretability, they can be prone to overfitting. Breiman [25] demonstrated that combining multiple independent, decorrelated decision trees to a random forest improves generalization performance.

To minimize the correlation between trees that are derived from the same training set, randomness can be injected during the learning phase. Therefore, two common randomization approaches are utilized. One method is to randomize the input data by drawing different bootstrap samples to grow different trees within the ensemble. Given a training set $\mathcal{D}_{train} = ((\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(m)}, y^{(m)}))$, a bootstrap sample is a subset $\mathcal{D}_B$ of that data, where each element was uniformly sampled with replacement from the original training set $\mathcal{D}_{train}$. Secondly, randomization can be injected during node optimization. Instead of splitting a node among all feature dimensions, only a random subset of all features can be considered for splitting in each tree. The idea is that by randomizing training of each tree, although individual trees might have high variance concerning particular subsets of the training data, the entire forest of trees will have a lower overall variance.

Drawing on the properties of decision trees, random forests provide a flexible framework with several degrees of freedom. Two important parameters are (1) the number of trees and (2) the maximum depth of each tree. A forest with an increasing amount of trees might be able to average out noisy predictions and decrease the error rate. Also, the maximum depth per tree is an important parameter as it influences the expressiveness of each tree. While the prediction of short trees might not be very confident as the leaf nodes might contain still a lot of heterogeneous samples, very deep trees might have too little data in the leaves and hence could overfit.

After training independent trees, the final predictions are made by aggregating the individual tree outputs. This is done in regression via averaging and in classification via majority voting.

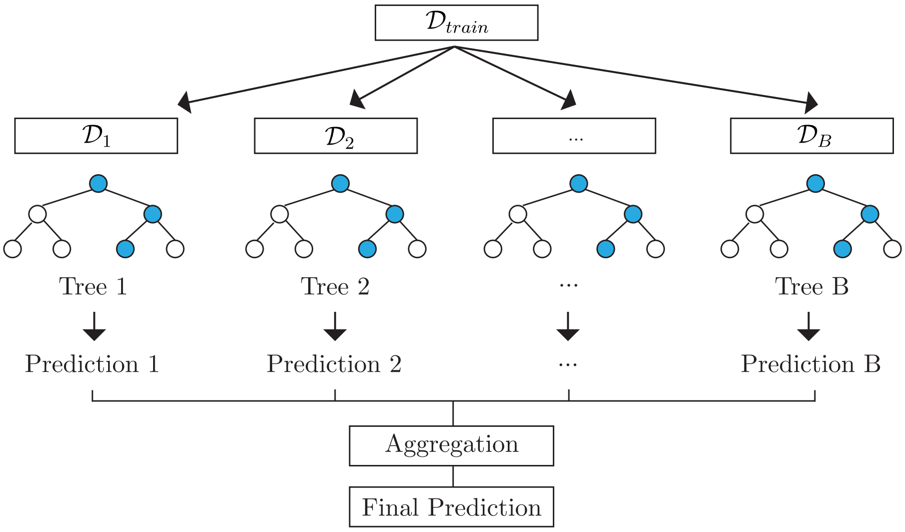


Figure 2.6: *Random Forest: B* trees are trained using different bootstrap samples. Subsequently, isolated predictions of single trees are aggregated to make a final prediction.

2.2 Reinforcement Learning

A key characteristic of supervised learning is that labeled training data is required to *supervise* the learning process. However, if we think about it, this is not the most natural way how e.g. humans learn. Consider an infant learning how to walk. Rather than having an explicit teacher that is tasked with the supervision, an infant more naturally learns to walk by interacting with its environment. Based on the feedback of the environment, e.g. cheering parents or falling on the floor, consequences of actions are learned in order to achieve some desirable goal. This is exactly the core idea of *reinforcement learning*, a paradigm that focuses on goal-directed learning from interactions.

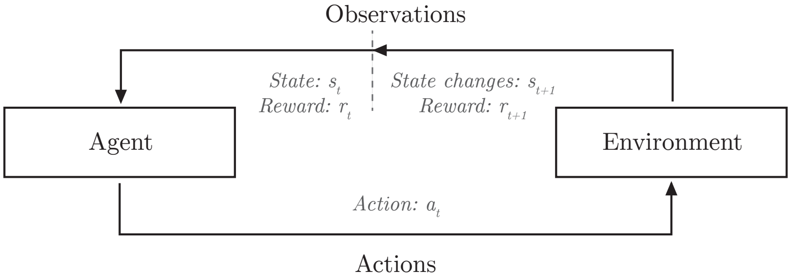


Figure 2.7: *Reinforcement Learning Setting:* An agent interacts with the environment via an action a_t and observes the environment in the next time step via s_{t+1} and r_{t+1} [29].

2.2.1 Markov Decision Process

At its core, reinforcement learning starts with a learner and decision maker called *agent*. The agent can perceive to a certain extent the outside world called *environment*. He can continually take *actions* on the environment that will result in a change of the environment's *state*. Jointly with the new state, the environment releases a scalar *reward* signal. The goal is to maximize the cumulative reward.

This sequential decision problem can be formalized as a so-called *Markov decision process* (MDP). In particular, we consider *finite* MDPs where the action and state space are finite. The basic elements of a finite MDP are introduced in the following based on [30, 31]:

- **State space** S: The state space is defined by the finite set of all possible states $\{s^{(1)}, ..., s^{(n)}\}$ of the environment. A *state* in this context refers to a unique representation of the environment's current situation.
- **Action space** A: The action space is defined by the finite set of all possible actions $\{a^{(1)}, ..., a^{(n)}\}$. Actions can be performed by the agent to control the environment's state. Generally, not all actions may be available in each state. In that case, $A(s)$ denotes the set applicable actions the agent can choose from in state s.
- **Transition function** T: The transition function $T(s, a, s')$ defines the transition of the environment from a state s to the next state s' based on a performed action a. The transition function can also be probabilistic in which case T defines a probability distribution over the next states:

$$T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t).$$

- **Reward function** R: The reward function implicitly specifies the learning goal and defines the scalar reward value $r \in \mathbb{R}$ an agent receives. There are two possible definitions for the reward function. The first option, $R : S \times A \mapsto \mathbb{R}$, rewards performing a specific action in a state while the second one, $R : S \times A \times S \mapsto \mathbb{R}$, rewards the transition from one state to the subsequent one. Both definitions are interchangeable.

At each discrete time step $t = 0, 1, 2, ...$ the agent receives a representation of the environment's current state $s_t \in S$ and selects an action $a_t \in A(s_t)$. Note that subscript

t denotes the state and the action chosen at time step t while the superscript above refers to uniquely defined states and actions of the environment. Based on an action, the environment transitions according to the transition function T into the next state s_{t+1} and further returns to the agent the reward $r_{t+1} \in R \subset \mathbb{R}$. Together the tuple (S, A, T, R) defines the finite MPD where the space of states and actions is limited to a finite number of instances.

2.2.2 Policy and Value Function

Policy Given an MDP, a policy is an agent's strategy that tells him which action to take in which situation. There are *deterministic* and *stochastic* policies. However, within the scope of this thesis we only consider deterministic policies. A deterministic policy π is defined as a function $\pi : S \mapsto A$. It returns for each possible state $s \in S$ one action $a \in A(s)$ to control the environment.

A desirable goal is to find the *optimal* policy. The notion of optimality is related to the rewards an agent gathers and can be different based on the time horizon of the MPD. For episodic tasks with a finite time horizon T, an optimal policy seeks to maximize the expected returns, the sum of future rewards, within this time frame:

$$E[\sum_{t=0}^{T} r_t]. \tag{2.21}$$

In the infinite horizon case, it is common practice to apply a discount factor $\gamma \in (0, 1)$ to future rewards, where forthcoming rewards are discounted more than earlier obtained rewards. This also leads to the convenient property that the infinite sum of rewards is finite:

$$E[\sum_{t=0}^{\infty} \gamma^t r_t]. \tag{2.22}$$

The discount factor can be considered as an interest rate. If the discount factor γ is 0, the agent is solely concerned about the immediate reward while a high discount factor takes future rewards into account more.

Value function One popular way to infer optimal policies is to learn a *value function.* A value function aims to quantify based on an optimality criterion how *good* it is for an agent to be in a certain state, or how *good* it is to take a specific action in a respective state. Value functions are defined on a policy level. Let $V^\pi(s)$ be the value of a state s under policy π, where the value can be interpreted as the expected returns of starting in s and following π afterward. Illustrative for the discounted, infinite-horizon model, the *state-value function* $V : S \mapsto \mathbb{R}$ can be defined as

$$V^\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s]. \tag{2.23}$$

Similarly, we can define the *state-action value function* $Q : S \times A \mapsto \mathbb{R}$ as the expected returns of choosing action a starting from state s and then following policy π:

$$Q^{\pi}(s,a) = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a]. \tag{2.24}$$

A fundamental property of every value function is that it satisfies a certain recursive property. More specifically, for any policy π and any state s, a value function can be recursively defined through the Bellmann equation [32], i.e.,

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... | s_t = s] \\ &= E_{\pi}[r_t + \gamma V^{\pi}(s_{t+1}) | s_t = s] \\ &= \sum_{s'} T(s, \pi(s), s')(R(s, \pi(s), s') + \gamma V^{\pi}(s')). \end{aligned} \tag{2.25}$$

This means that the expected value of a state can be decomposed in the immediate reward and the discounted value of next possible states weighted by their transition probabilities $T(s, \pi(s), s') = P(s'|s, \pi(s))$. Analogously, this also holds for the state-action value function.

The key objective in reinforcement learning is to find an optimal policy π^* such that $V^{\pi^*}(s) \geq V^{\pi}(s)$ for all $s \in S$ and all policies π. Note that multiple different policies could possibly lead to the same optimal state-value function. The optimal solution $V^* = V^{\pi^*}$ can be defined as

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s,a,s')(R(s,a,s') + \gamma V^*(s')). \tag{2.26}$$

This means that the state-value under an optimal policy is equal to the expected return of taking the best action in that state. Given an optimal state-value function, the optimal action can be greedily chosen according to

$$\pi^*(s) = \arg \max_{a \in A(s)} \sum_{s' \in S} T(s,a,s')(R(s,a,s') + \gamma V^*(s')). \tag{2.27}$$

In the same vein, the optimal state-action value function Q^* can be defined as follows,

$$Q^*(s,a) = \sum_{s' \in S} T(s,a,s')(R(s,a,s') + \gamma \max_{a'} Q^*(s',a')). \tag{2.28}$$

As $V^*(s)$ represents the maximum expected returns when starting from s, it is equal to the action that results in the maximum Q-value from s. More formally, the following relationship holds:

$$V^*(s) = \max_{a} Q^*(s,a). \tag{2.29}$$

This, in turn, means that choosing the action with the maximum expected future returns is equivalent to taking the action with the maximum optimal state-action value. The optimal action π^* according to a greedy policy results to

$$\pi^*(s) = \arg \max_{a \in A(s)} Q^*(s,a). \tag{2.30}$$

[30]

2.2.3 Q-Learning and DQN

Using the Q-function instead of the V-function has the convenient property that no prior knowledge about the transition and reward function is required. This makes Q-functions especially applicable in *model-free* approaches, where the model of the MDP is not known beforehand. One approach to derive an optimal policy is to estimate the Q-function for different actions based on the reward signal and act according to equation 2.30. Naturally, an agent faces an *exploration-exploitation* trade-off, where the agent has to balance exploring the MDP to gather more information versus selecting high-value actions based on the so-far obtained experience [30].

Q-Learning A basic approach to estimate the Q-values is the Q-learning algorithm. Q-learning in its vanilla form [33] aims to estimate the Q-values by iteratively updating the values for each state-action combination within a tabular representation until convergence. State-action values are incrementally updated according to the following update rule

$$Q_{i+1}(s_t, a_t) = Q_i(s_t, a_t) + \alpha \left(R(s_t, a_t, s_{t+1}) + \gamma \max_a Q_i(s_{t+1}, a) - Q_i(s_t, a_t) \right) \tag{2.31}$$

where a_t is an action chosen according to an exploration strategy, α is the learning rate and γ denotes the discount factor. Note that regardless of the exploration strategy Q-learning will eventually converge assuming that α is decreased appropriately and each state-action pair is visited an infinite number of times. However, for MDPs with large action and state spaces, the resulting exploratory size of state-action combinations make vanilla, tabular Q-learning memory-inefficient and computationally intractable [30].

Deep Q-learning (DQN) In order to overcome the mentioned limitations of tabular Q-learning, [34] uses instead a deep neural network as a non-linear function approximation to estimate the Q-function. The *deep Q-Network (DQN)* is a deep neural network that takes the state s as an input and outputs estimated Q-values for each available action $a \in A(s)$ by performing a forward pass. The Q-Network parametrized by θ is trained by minimizing the following loss function

$$L_i(\theta_i) = E_{s,a \sim p(.)}[(y_i - Q(s_t, a; \theta_i))^2] \tag{2.32}$$

where $y_i = E_{s'}[r + \gamma \max_{a'} Q(s', a'; \theta^-)|s, a]$ denotes the target in iteration i and $p(s, a)$

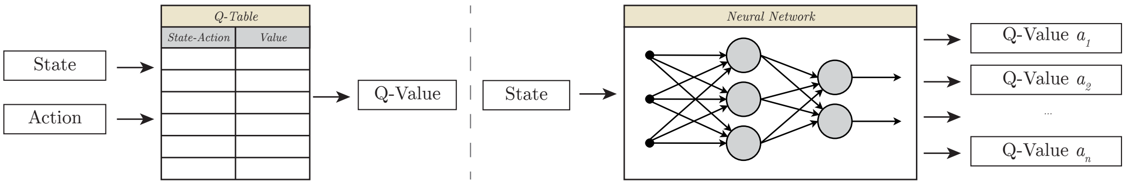


Figure 2.8: *Left*: Q-Learning keeps track of Q-values in a tabular representation for each possible state-action pair. *Right:* In Deep Q-Learning a DQN estimates the Q-values for each action available from a given state.

represents a distribution over states and actions. Note that for the target estimation of y_i a second separate network with identical structure but different weights θ^- from a previous iteration is used. Fixing the parameters θ^- of the target network improves stability while optimizing the loss function. The parameters of the target network can be updated every τ time steps by copying the parameter of the Q-Network $\theta^- \leftarrow \theta_i$. The respective loss function can be optimized using stochastic gradient descent by differentiating the loss with respect to the weights:

$$\nabla\theta_i L_i(\theta_i) = E_{s,a\sim p(.)}[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s_t, a; \theta_i))^2 \nabla\theta_i Q(s, a; \theta_i)]. \quad (2.33)$$

Furthermore, the authors utilize an experience replay memory where the agent's experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ at each time step t are stored in a memory. During the learning process, experiences are sampled from the replay memory for the agent in order to break the correlation between consecutive samples and reduce the variance of the weight updates [34]. Since its initial introduction, many improvements drawing on the original DQN algorithm have been proposed. Since Q-values are very noisy, taking the max over all action will likely return an overestimated value. Double DQN [35, 36] tackles the problem of the overestimation of Q-values. The authors suggest using two different Q-networks, an online network parametrized by θ that chooses the next action and one target network parametrized by θ^- that estimates the Q-values. This will lead to the update $y_i = E_{s'}[r + \gamma Q(s', \arg\max_a Q(s', a, \theta_i); \theta_i^-)|s, a]$. By decoupling the policy from its evaluation, the authors show reduced overestimation and improved performance. Building upon double DQN, [37] introduced prioritized experience replay. The intuition is to increase the replay probability of experience samples that have high expected learning progress to accelerate the learning process. Dueling DQN [38] proposes a new neural network architecture called *dueling network*. They break down the Q-value estimation in two components, one that estimates how good it is to be in a state $V(s)$ and another one that estimates the *advantage* $A(s, a)$ of taking a corresponding action in that state. This is realized by two separate streams within their network architecture, which are later combined to estimate the Q-values. This enables to separately learn the state-value function, without being influenced by the advantage of taken actions.

2.3 Active Feature-Value Acquisition (AFA)

A common assumption in many traditional machine learning problems is that training data is readily available and complete for training models. However, in practice, data is not always complete. Instead, further information (features) can be acquired from different sources at some processing or acquisition cost until a certain budget is exhausted. Consider a motivating example where a patient at the hospital is looking for diagnosis and treatment of his current condition. Initially, a doctor will have insufficient information to recommend an appropriate treatment. First, a patient has to complete a survey about their medical history at the reception. Next, the doctor assesses the patient's symptoms within a consultation hour. However, a confident diagnosis is often not possible after the first consultation. So subsequently, further information has to be acquired through e.g. procuring lab tests or getting specialist opinions. At each step, some medical or administrative cost occurs. Clearly, conducting all possible medical tests initially at once, meaning having all features readily available, is monetary and time-wise not feasible. Rather, a doctor aims to efficiently acquire the next feature that will decrease uncertainty until he can provide a confident diagnosis. This setting is called active feature-value acquisition (AFA) and spans beyond medical domains to online advertisement, credit assessment, disaster mapping and more [39, 40, 41].

2.3.1 Traditional Prediction-time AFA

Different than *induction-time* AFA [10], in *prediction-time* AFA [41] querying new features takes place at test time. In particular, a model is trained on complete training instances and sequentially queries feature values per instance at test time in order to trade off cost and predictive performance.

Let h be a classifier induced from a complete training set of d features and their corresponding ground-truth labels. Next, a test set with m samples is given, where each instance is also represented by d feature values, which can be initially unobserved. At time step $t = 0$, each of the m test examples $\mathbf{x}^{(i)}$ start with an initial empty set of features $\mathcal{O}_0^{(i)} = \{\}$. At each time step t , additional feature values $\{x_j^{(i)}\}_{j \in \mathcal{S}_t^{(i)}}$ from the set of unselected features $\mathcal{S}_t^{(i)} \subseteq \{1, \ldots, d\} \setminus \mathcal{O}_{t-1}^{(i)}$ can be acquired at the cost $c_t^{(i)} = \sum_{j \in \mathcal{S}_t^{(i)}} c_j$. These features are determined according to a feature acquisition strategy. After each feature collection step, the predictor has access to the features in $\mathcal{O}_t^{(i)} := \mathcal{S}_t^{(i)} \cup \mathcal{O}_{t-1}^{(i)}$. The acquisition process continues until a stopping criterion is met at $t = T$, where the classifier makes the prediction of $h(\mathbf{x}^{(i)})$ based on the partially observed features in $\mathcal{O}_T^{(i)}$. Generally, the desirable goal is to carefully query features on an instance-level that result in the best prediction performance at the lowest cost across the test set. Note that different individuals within the test set can have personalized sets of features with different observed features.

Naturally, this traditional prediction-time AFA setting needs to incorporate three elements: (1) a mechanism for the classifier to handle missing feature values during prediction time, (2) a feature acquisition strategy that subsequently determines the next feature to query per instance and (3) a stopping criterion which stops the acquisition process and predicts the outcome of a test instance based on the queried features.

Handling missing values There are two common approaches for dealing with missing data during prediction time: imputation-based methods and reduced feature models. The main idea of imputation-based approaches is to estimate a missing value or its distribution based on available data. Predictive value imputation (PVI) replaces missing values with their estimates before a prediction is made by the model. The estimation method may range from simple ones such as the mean or mode of a feature to more rigorous predictive models. Distribution-based imputation uses the training data to estimate a distribution over the values of an attribute in order to compute the expected distribution of the target variable. Estimated probabilities of the missing values can then be used to weight the prediction outcomes. This is a commonly used strategy for tree-based classifiers [42]. For example, [43] introduces a probabilistic random forest, that treats features and

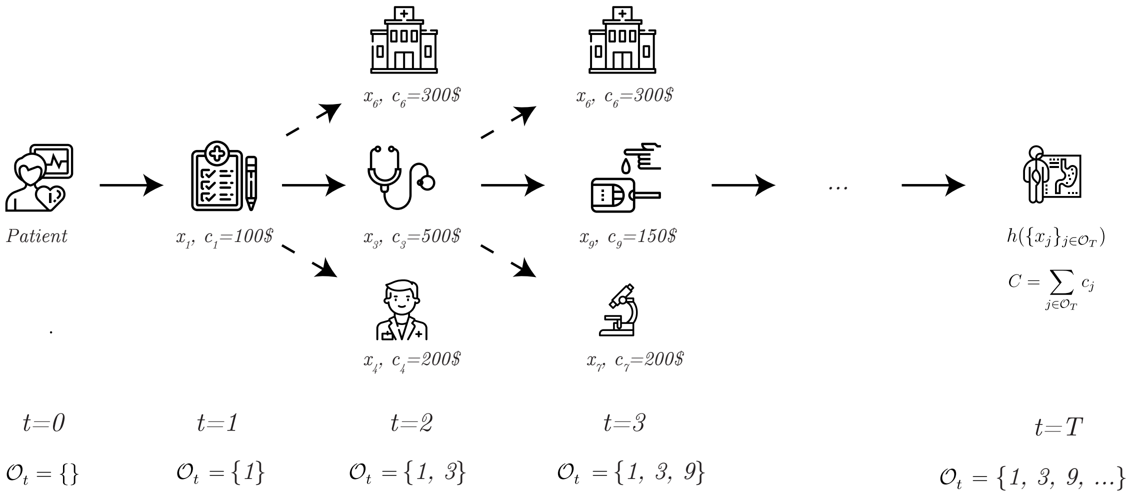


Figure 2.9: *Prediction-time AFA:* At test time features values x_i can be sequentially queried at cost c_i. At time step T, when a stopping criterion is met, a prediction is based on the partially observed feature set.

labels as probability distribution functions, and naturally copes with missing values. An alternative approach to imputation-based methods are reduced feature models. Reduced feature models apply multiple different models that were induced only on a subset of features. If at test time a partially missing feature vector is observed, the model that has been trained only on that particular subset of observed features is employed for prediction. However, the number of models to be trained and stored is exponential in the number of attributes, which makes this method storage-wise expensive [42]. A different line of work makes use of set encoders [44, 45, 46] to embed partial feature sets in a representation that can be further processed by neural networks.

Feature acquisition strategy Different heuristic-based approaches have been adopted from the active learning [47] and traditional AFA literature [10, 48] to select the next relevant feature on an instance-level. Uncertainty sampling is based on the idea that prediction errors mainly occur when predictions are ambiguous and no confident decision can be made - in other words when a lot of uncertainty is involved. For example, missing feature values can be estimated by imputation and the imputed value the model is least certain about can be queried. Uncertainty can be measured e.g. by unlabeled margins, which represents a model's ability to differentiate between instances of different classes. An alternative approach is to estimate the expected improvement in utility for each feature query. This could be e.g. the expected marginal contribution to the predictive performance [10, 48, 41]. More traditional feature selection methods such as L1 regularization for linear classifiers [49] or feature importance rankings e.g. by a random forest [25] also can choose effectively a subset of features. However, they focus on selecting a fixed, static set of features for the whole data set instead of personalizing features to individuals. A more recent framework called EDDI [46] uses an acquisition function inspired by a Bayesian experimental design that selects features to maximize the expected information gain given the set of already observed features.

Stopping criteria Based on an initially empty set of features, an AFA system iteratively collects features that have not been observed yet until a certain stopping criterion is met. One method is to stop when a desirable level of performance is reached. However, most prior work on AFA does not explicitly state a stopping criterion, but rather the acquisition process stops when a certain global budget, group budget or individual-level budget is exhausted [11, 50].

2.3.2 RL-based AFA

While the three components within traditional AFA systems are mostly independent and disentangled, another approach in the literature [45, 50, 51] formulates the prediction-time AFA setup as a reinforcement learning problem where i.e. the feature acquisition strategy and the stopping criterion are intertwined within a unified framework. The prediction-time AFA setting can be represented as a Markov decision process (MDP), where an agent chooses in each state an action according to its policy with the goal to maximize rewards. The MDP can be defined as the following:

- **State**: At each time step t the state $s_t^{(i)} = \mathcal{O}_t^{(i)}$ is represented by the currently observed set of selected features of individual / episode i. The state space S is defined by the powerset of the feature set, which has a cardinality of 2^d.
- **Action**: The set of possible actions $A(s_t^{(i)})$ at time step t is defined by the set of unselected features $\mathcal{S}_t^{(i)} \subseteq \{1, \ldots, d\} \setminus \mathcal{O}_{t-1}^{(i)}$ for individual i combined with an additional $\{STOP\}$ action, that terminates the episode and distributes the final reward to the agent.
- **Reward**: Most prior work [45, 51] proposes a reward function that balances costs with predictive performance using a trade-off parameter λ. At each time step t, the agent is penalized with the cost of the acquired feature. However, in the case of the terminal classification action, wrong classification is penalized using a suitable loss function, e.g.,

$$r_t = R(\mathcal{O}_t^{(i)}) = \begin{cases} -\lambda c_j & \text{if } a = j \\ -l(h(\mathcal{O}_t^{(i)}), y^{(i)}) & \text{if } a = \text{STOP}. \end{cases} \tag{2.34}$$

Translating the problem into a general MDP opens the solution space to reinforcement learning methods such as deep Q-learning in order to learn an optimal policy that maximizes the expected cumulative rewards $E[\sum_{t=0}^{T} r_t]$ and hence trades off the acquisition costs with accurate predictions. Importantly, the policy jointly addresses both the feature acquisition strategy and the stopping criterion within the action space.

2.4 Fairness in Machine Learning

Machine learning systems are nowadays integrated into our daily life, may it be filtering out spam emails or unlocking our phones using facial recognition software. More and more of these systems are also employed in high-stake situations such as credit risk assessment [52], healthcare [53] or criminal justice [54], in order to support or automate decision making. While being accurate, numerous studies have revealed that these systems can unintentionally encode biases and introduce systematic discrimination against minority groups within the population [7, 5, 55]. A prominent example is the COMPAS algorithm, which has been employed in the US criminal justice system to predict the risk of defendants to recommit a crime. A study from ProPublica [5] found that these predictions show a higher false positive rate for black than for white defendants. Similar studies showed biases against gender and race in commercial facial recognition software [7] and recruiting systems [55]. Consequently, the academic community became increasingly interested in studying fairness in machine learning and exploring methods to mitigate discrimination.

2.4.1 Sources of Unfairness

Inequalities and discrimination in machine learning systems often relate to bias. Frequently, the term bias in this context refers to demographic disparities that are worrisome for societal reasons. These disparities can be introduced in different stages of the machine learning loop. According to [56], the typical stages within a machine learning system that outputs predictions comprise *measurement*, *learning*, *action* and *feedback*.

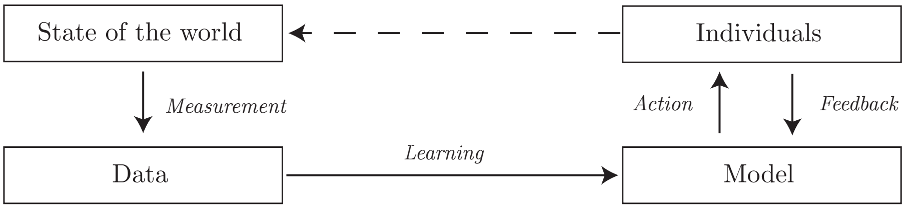


Figure 2.10: The (simplified) machine learning loop [56].

Measurement In the first stage, the current state of the world is measured and represented as data sets. Often, these collected data sets will likely also encode biases that also exist in our society. This could be due to explicit historical discrimination, implicit societal stereotypes or distributional differences of certain attributes. For instance, some occupations have a very high gender imbalance. An automated job screening application for technical positions trained on this imbalanced data set could be prone to discriminate against one gender.

Learning In the learning stage, a machine learning model is inferred using the data measured at the first stage. Intuitively, learning models from biased data sets without interventions will also likely lead to biased predictions. However, disparities can also be introduced when they are not contained in the training data. The most prevalent reason for this is the sample size disparity. Naturally, there are fewer data points about minorities when the training set is sampled uniformly from the population. Training the model on less data about minorities generally means that predictions tend to be more erroneous for minorities than for the general population.

Action The next stage leverages outputs of a model on unseen examples to take actions, such as deleting a spam mail or detaining a criminal. However, even if models do not introduce disparities in the previous stages, it can become a problem when characteristics in the population shift, a phenomenon known as *distribution shift*. If different sub-groups within the population change differently over time, this can lead to disparities.

Feedback In some scenarios, algorithmic systems receive feedback after making predictions. For example, major cities use predictive policing systems to forecast criminal hotspots and allocate police to those high-risk areas based on crime data. The feedback signal, the crimes discovered in those areas, is subsequently used to update and refine the model. This can bias the model further towards those neighborhoods. However, it is debatable if crimes were recorded due to an actual higher crime rate in those claimed high-risk areas or because of increased scrutiny of police sent to those neighborhoods. These feedback loops can lead to self-fulfilling predictions where police are repeatedly sent to the same areas regardless of the true crime rate.

2.4.2 Notions of Statistical Fairness

While it is rather intuitive to develop a sense of discrimination, it is not obvious what it means for a system to be *fair*. The literature mainly proposed two families of fairness notions in algorithmic decision systems: *individual* and *statistical* notions of fairness. Individual notions of fairness try to guarantee fairness on an individual level such that "similar individuals are treated similarly" [57]. However, this requires a task-specific similarity metric, which is difficult to agree upon in practice. Statistical notions of fairness (sometimes called *group fairness*) on the other hand, require some statistic of a classifier to equally hold across some defined protected subgroups. While this family of definitions is easy to measure, it fails to provide guarantees to individuals.

Let, as previously introduced, individual i be represented by an n-dimensional feature vector $\mathbf{x}^{(i)}$ and its corresponding binary outcome label $y^{(i)}$. In addition, let $b^{(i)} \in \{0,1\}$ be a binary *sensitive* or *protected* attribute, which indicates to which population subgroup an individual belongs to. Further, $(\mathbf{x}^{(i)}, y^{(i)}, b^{(i)})$ are drawn from the joint probability distribution $\mathcal{P}$. The idea of protected groups is legally rooted in anti-discrimination laws. Importantly, just omitting the sensitive attribute from the data set and trying to achieve *fairness through unawareness* can still lead to discrimination, as the sensitive attribute might be redundantly encoded in other features (proxies) [57, 58].

A plethora of fairness definitions have been introduced in recent years and [59] enumerated as many as 20 different definitions. In this work, we focus on the family of statistical fairness in a binary one-shot classification setting. In the following, we introduce four popular definitions of statistical fairness. Let's consider an example where gender is the sensitive attribute and a classifier h predicts if a loan applicant is creditworthy or not.

Demographic Parity Demographic parity or sometimes called statistical parity requires the predictions to be independent of the sensitive attribute. The idea is that each group has an equal chance of receiving a positive outcome. In other words, the loan acceptance rate should be equal across male and female applicants [57].

$$P(y = 1 | b = 0) = P(y = 1 | b = 1) \tag{2.35}$$

Equal Odds A predictor satisfies the criterion equalized odds if *both* the false negative rate (FNR) *and* the false positive rate (FPR) are equal across the subgroups. In contrast to demographic parity, equalized odds takes the ground-truth outcome of an individual in addition to their sensitive group into account. In the illustrated example, this means that the probability of a creditworthy applicant to be correctly classified is equal across groups and also the probability of an unqualified applicant to be incorrectly provided a loan [60].

$$P(h(\mathbf{x}) = 1|y = z, b = 0) = P(h(\mathbf{x}) = 1|y = z, b = 1) \qquad z \in \{0, 1\} \tag{2.36}$$

Equal Opportunity In some domains, we just care about the inclusion error. Equal opportunity is a relaxed, weaker notion of equal odds that just requires only the TPR of both subgroup to be the same. In our example, the probability of being accepted a loan while being creditworthy should be equal for the male and female subgroup [60].

$$P(h(\mathbf{x}) = 1|y = 1, b = 0) = P(h(\mathbf{x}) = 1|y = 1, b = 1) \tag{2.37}$$

Calibration by Group In some applications, it is desirable to interpret the values of a scoring classifier as (calibrated) probabilities. The calibration property introduced in Section 2.1.3 holds if for all individuals with a particular assigned probability score $\pi \in [0, 1]$, a π-fraction of them actually belongs to the positive class. A probabilistic classifier satisfies calibration by groups if this condition holds equally for each subgroup. For example, if within the population each 100 male and female applicants receive a probability score of $\pi(\mathbf{x})$ =0.7 of being creditworthy, 70 individuals of them belong actually to the positive ground-truth class [61].

$$P(y = 1|\pi(\mathbf{x}) = p, b = z) = p \qquad z \in \{0, 1\} \tag{2.38}$$

It turns out that the introduced statistical fairness definitions can be at odds with each other and are mutually exclusive. In particular, the impossibility theorem says that except in trivial settings (e.g., equal base rate or perfect classifier), it is impossible to simultaneously achieve demographic parity, equal odds and calibration by groups [56, 54, 62, 63].

2.4.3 Achieving Statistical Fairness

The fairness literature proposed many algorithmic interventions to achieve statistical notions of fairness. Most approaches to promote fairness fall into three categories: pre-processing, in-processing, and post-processing techniques. Using these techniques to decrease the degree of discrimination comes often at the cost of predictive performance.

Pre-processing This family of techniques preprocesses a data set to remove discrimination prior to learning a classifier. Hence, this approach is generally agnostic to potential downstream predictions. Most preprocessing methods aim for demographic parity by transforming the feature space into some representation that is independent of the sensitive attribute. A straight-forward approach is *suppression*, which suggests simply removing features that are correlated with the sensitive attribute. However, this might be ineffective if the relationship between features and the sensitive attribute is not linear but more complex. *Massaging the data* aims at mitigating demographic disparity by changing the labels of some individuals in both groups in the data set and use a ranker to pick candidates that are close to the decision boundary to motivate minimal effects on accuracy. As changing the labels is a rather intrusive approach, the authors also suggested reweighing and sampling techniques to correct for sample size disparity across the intersection of class labels and protected groups [64]. The *optimized preprocessing* framework learns a probabilistic transformation that transforms the features and outcomes within the data using group fairness, individual distortion, and data fidelity constraints [65]. A further line of work aims to learn a fair representation from the data using adversarial learning. The high-level idea is that a discriminator aims to predict the sensitive attribute while an encoder aims to map each data distribution to a single representation to fool the discriminator. While earlier work [66, 67] achieved representation ensuring demographic parity, [68] extends this approach to equal opportunity and equal odds by choosing different loss functions.

In-processing Fair in-processing methods aim at mitigating disparities during the training process. While optimizing directly for fairness is highly effective, the disadvantage is limited generalization as this applies only to specific model classes and optimization problems. One line of work introduces fairness criteria as a constraint to a constrained optimization problem. Zafer et al [69] formulate an optimization problem which given a decision boundary-based classifier (e.g. logistic regression or support vector machines) minimizes the loss subject to a fairness constraint. In order to solve this problem efficiently, they convert it into a Disciplined Convex-Concave Program [70]. They satisfy different notions of fairness regarding (approximate) equal overall misclassification rate, false positive rate and false negative rate across subgroups. Contrary to their previous approach, the same authors also propose to maximize fairness under accuracy constraints instead of vice versa [71]. Agarwal et al [72] propose an approach where they reduce a constrained optimization problem with fairness constraints to a sequence of cost-sensitive classification problems with two players. At each round of the sequence, one player maximizes accuracy while the other player imposes a particular amount of fairness. The solution of this cost-sensitive classification problem yields a randomized classifier with the lowest error while satisfying different fairness definitions such as equal odds, equal opportunity, and demographic parity. In [73], a meta-algorithm for classification is proposed that takes a large class of fairness constraints with respect to possibly multiple non-disjoint sensitive attributes as input and optimizes the classifier subject to the fairness constraints.

Post-processing After a classifier is trained on a data set, post-processing approaches take the model's predictions and adjust them in order to aim for fairness. The advantage of this class of methods is that it is model-agnostic and works for any black-box classifier. Further, there is no need to have access to the training process in contrast to in-processing methods. A popular method to achieve calibration is Platt scaling [74]. Platt scaling treats an uncalibrated score s as a single feature and fits a logistic regression model such that the parameter a and b fit the sigmoid function $\pi = \frac{1}{1+exp(as+b)}$. Subsequently, this regression model can be used to transform scores into probabilities. To get calibration by group, Platt scaling can be simply applied separately to the different subgroups. Hardt et al [60] introduce a post-processing technique that achieves equal odds by using randomized, group-specific thresholds to make class predictions. They use a mixture of two thresholds t_{low} and t_{up} per group, where probability scores above $\pi(\mathbf{x}) > t_{up}$ are always classified as 1 and scores $\pi(\mathbf{x}) < t_{low}$ are always classified as 0. For scores in between however, i.e. $t_{low} < \pi(\mathbf{x}) < t_{up}$, the classifier will flip a coin to assign the class. By having a fraction of randomized predictions per group, this method can tune the error rates for each group in order to equalize both $FPR \ and \ FNR$. Pleiss et al [75] also rely on randomization to post-process existing calibrated classifiers to achieve single error parity (e.g. equal opportunity) in addition to calibration. In contrast to the previous method, they do not randomize the output but rather assign the base rate probability to a fraction α of individuals of the advantaged group. In other words, they mix the trivial classifier that always outputs the group-specific mean probability with the previously learned and calibrated classifier in order to tune the error rate while preserving the calibration property. The work most related to the one presented in chapter 3 and 4 is the *active fairness* framework [11], which also investigates fairness in prediction-time AFA systems. Their post-processing technique is based on acquiring different amounts of features (information budgets) for different subgroups. This additional degree of freedom allows it to control for error rates as more features for a group will improve the predictive performance. First, they achieve calibration and single error parity by choosing differentiated group budgets jointly with a calibrated classifier. Second, they combine group-specific information budgets with different classifier thresholds and achieve hereby equal odds. In order to find the right set of parameters (budget or threshold) that leads to fair classification, they rely on optimization methods. This work is different from theirs in that we propose two alternative approaches to determine information budgets that reduce disparities. Crucially, our approach determines these information budgets dynamically at prediction-time when a stopping criterion is met and does not rely on optimization. The first approach adaptively acquires features for an individual until a certain level of confidence in a prediction is reached. The second approach relies on a reinforcement learning agent that collects features for an individual and further finalizes the information budget by deciding when to stop acquiring additional features.

Chapter 3

On Fairness in Budget-Constrained Decision Making

Contributing Article: Bakker, A. M., Noeriega-Campero, A., Tu, D. P., Sattigeri, P., Varshney, K., Pentland, A. S. On Fairness in Budget-Constrained Decision Making. *Workshop on Explainable AI (XAI) for Fairness, Accountability, Transparency at KDD 2019.*

Declaration of Contributions

	Bakker, Michiel	Noeriega-Campero, Alejandro	**Tu, Duy Patrick**	Sattigeri, Prasanna	Varshney, Kush	Pentland, Alex "Sandy"
Development of idea	x	x				
Development of theory	x					
Provided helpful discussions and input	x	x	**x**	x	x	
Implementation of framework	x		**x**			
Data processing and experiments			**x**			
Analysis of results			**x**			
Writing of the manuscript	x		**x**			
Improvement of text	x	x	**x**		x	
Supervision of research					x	x
Presentation of research	x		**x**			

On Fairness in Budget-Constrained Decision Making

Michiel A. Bakker
MIT-IBM Watson AI Lab
MIT Media Lab
Cambridge, MA
bakker@mit.edu

Alejandro Noriega-Campero
MIT-IBM Watson AI Lab
MIT Media Lab
Cambridge, MA
noriega@mit.edu

Duy Patrick Tu*
MIT-IBM Watson AI Lab
MIT Media Lab
Cambridge, MA
patrick2@mit.edu

Prasanna Sattigeri
MIT-IBM Watson AI Lab
IBM Research
Yorktown Heights, NY
psattig@us.ibm.com

Kush R. Varshney
MIT-IBM Watson AI Lab
IBM Research
Yorktown Heights, NY
krvarshn@us.ibm.com

Alex 'Sandy' Pentland
MIT-IBM Watson AI Lab
MIT Media Lab
Cambridge, MA
pentland@mit.edu

ABSTRACT

The machine learning community and society at large have become increasingly concerned with discrimination and bias in data-driven decision making systems. This has led to a dramatic increase in academic and popular interest in algorithmic fairness. In this work, we focus on fairness in budget-constrained decision making, where the goal is to acquire information (features) one-by-one for each individual to achieve maximum classification performance in a cost-effective way. We provide a framework for choosing a set of stopping criteria that ensures that a probabilistic classifier achieves a single error parity (e.g. *equal opportunity*) and calibration. Our framework scales efficiently to multiple protected attributes and is not susceptible to intra-group unfairness. Finally, using one synthetic and two public datasets, we confirm the effectiveness of our framework and investigate its limitations.

KEYWORDS

algorithmic fairness, equal opportunity, active feature acquisition, budget-constrained decision making

ACM Reference Format:
Michiel A. Bakker, Alejandro Noriega-Campero, Duy Patrick Tu, Prasanna Sattigeri, Kush R. Varshney, and Alex 'Sandy' Pentland. 2019. On Fairness in Budget-Constrained Decision Making. In *Proceedings of KDD '19: Explainable AI (XAI) for Fairness, Accountability, Transparency (KDD '19: Explainable AI (XAI))*. ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

As machine learning-based decision making has become increasingly ubiquitous—e.g., in criminal justice [16], medical diagnosis [15], human resource management [3], credit [11], and insurance [29]—there is widespread concern over how these systems introduce and perpetuate discrimination and inequality. Consequently, substantial work on defining and achieving fairness in machine learning systems has been published in the last few years.

The vast majority of this research has relied on the assumption that all data is readily available or can be acquired at no or little additional costs. In such a setting, the model bases its decision about an individual always on all features. In practice, however, there are many applications where the acquisition of an additional feature leads to a feature-specific cost [17]. Consider a patient entering a hospital seeking diagnosis. Typically, the doctor starts the diagnosis with only a handful of symptoms. From there, the patient undergoes a progressive inquiry by e.g. measuring vitals or procuring lab tests. At each step, absent sufficient certainty, the inquiry continues. Acquiring all features at once using all possible medical tests is prohibitively expensive and time-consuming, so at each time-step the doctor is tasked with choosing the next feature that most efficiently leads to a more confident diagnosis. This setting, *active feature-value acquisition* (AFA), is becoming increasingly ubiquitous and is relevant in a wide range of contexts, from credit and insurance, to employee recruiting, poverty and disaster mapping, and online advertising [8, 17, 19, 22, 28].

The machine learning community has proposed different frameworks for quantifying fairness in machine learning [10, 16, 30], most of which focus on balancing classification errors across protected population subgroups, towards achieving equal false-positive rates (*predictive equality*), equal false-negative rates (*equal opportunity*), or both (*equal odds*). Here, we focus on satisfying equal opportunity, requiring non-discrimination only within the 'favorable' outcome [10], while also extending these results to satisfying predictive equality. We show that our method can jointly achieve either of these error parity measures and calibration for each subgroup (*test-fairness*), a property commonly required of classifiers in real-world settings [5, 25]. We call an estimator *calibrated* if, when we look at the subset of people who receive any given probability estimate $p \in [0, 1]$, we indeed find a p fraction of them to be positive instances of the classification problem.

To ensure that predictions are fair, “optimal" post-processing methods have been proposed that achieve either 1) equal odds, or 2) parity in one error rate (e.g. equal opportunity) and calibration [10, 25]. These methods rely on randomization to attain fairness:

[*] Duy Patrick Tu did this work while visiting MIT from LMU Munich.

KDD '19: Explainable AI (XAI), August 4–8, 2019, Anchorage, AK

they randomize the predictions for a subset of individuals in the advantaged group, and hence increase error rates for that group. By carefully tuning the share of randomized predictions, one ensures equal error rates across groups. Although these methods are effective, they are also unsettling and several objections to them have been put forth, such as inefficiency, pareto-suboptimality, and intra-group unfairness due to the randomization [5, 10, 20, 25].

Despite the pervasiveness of AFA systems and the recent spike in work on algorithmic fairness, only one paper in the literature has explored fairness at its intersection with AFA [22]. In that work, optimization is used to find an information budget for each population subgroup such that an AFA classifier achieves either 1) one error parity and calibration or 2) equal odds. Notably, by using this additional degree of freedom, they show that one can achieve these notions of fairness in an AFA setting without resorting to randomization.

Our goal is to further investigate the relationship between equalizing error rates and AFA. In particular, we derive a set of stopping criteria that ensures single error parity (equal opportunity or predictive equality) for calibrated probabilistic classifiers. In contrast to previous work, this method does not rely on optimization but directly relates the stopping criteria to the subgroup-specific base rate and the desired error rate. We demonstrate that our framework is effective in practice using one synthetic and two public datasets and show how it extends naturally to a situation with many subgroups defined over multiple protected attributes.

Finally, the method provides an interesting new perspective on two central topics in the fairness literature: *individual fairness* and *fairness gerrymandering*. First, as statistical notions of fairness like equal opportunity are defined with respect to groups, they only provide guarantees to the group average, not to any individual. Individual fairness tries to tackle this issue by using constraints that bind at the individual level [7]. Our method finds a set of stopping criteria that lead to a personalized budget and set of available features for each individual. Hence, intuitively, it trades off inequality (the model and the budgets are personalized and thus different across individuals) for equity (each of the subgroups defined over the set of protected attributes has the same expected false-negative rate and even within subgroups each individual is classified with similar confidence). This can be seen as an attempt to combine statistical and individual notions of fairness as the stopping criteria lead to increased equity at the individual level. Second, in *fairness gerrymandering*, a classifier appears to be fair when measured across each protected attribute but violates the fairness constraint on a subgroup defined over several protected attributes [14]. In contrast to methods based on optimization, our framework is robust against fairness gerrymandering since it ensures all subgroups have the same expected false-negative rate.

2 RELATED WORK

Active feature-value acquisition. Several methods for AFA have been explored ranging from heuristics-based feature acquisition strategies to more recent reinforcement learning methods in which one jointly trains the classifier and the agent that decides which feature to select next [8, 17, 19, 28]. In line with most prior work in AFA, we select the next best feature using a feature acquisition strategy while separately training the classifier. The feature acquisition strategy is based on maximizing the expected utility

$$EU(x_j) = \int_v P(x_j = v)\frac{U(x_j = v)}{c_j} \tag{1}$$

where $P(x_j = v)$ is the probability that feature x_j will take on value v and $U(x_j = v)$ the utility of the model after adding x_j to feature vector $\mathbf{x}$. The utility function could be defined in multiple ways depending on the objective, such as the expected classification error or the expected entropy. We experimented with multiple definitions for utility and found that one that maximizes the expected increase or decrease in probability outputted by the model is most cost-efficient; see Section 5.1 for details.

Fairness in machine learning. Most recent work in fairness in machine learning, including this work, focuses on matching error rates (false-positive or false-negative) across population subgroups. There are, however, multiple other ways to define fairness such as *demographic parity*, *individual fairness*, *fairness through unawareness*, and *counterfactual fairness*. Please refer to [30] for a comprehensive overview of definitions. Methods for achieving fairness fall into three categories [1]. First, there are methods for pre-processing and improving collection of training data [2, 4, 27]. Second, there are methods for constraining the model during training or optimization including methods for fair representation learning [21, 33]. Finally, there are a number of methods for post-processing probabilities to achieve fairness [10, 32]. For achieving equal opportunity and calibration previous post-processing work has relied on randomization which led to an inefficient and pareto-suboptimal classifier [5, 25]. In this work, we post-process a classifier trained on all features by selecting a specific subset of features for each individual.

3 PROBLEM SETUP

The setup of our framework most follows the one in [25] for fairness in the context of calibrated probabilistic classifiers. However, we extend their framework for use in the AFA setting. Let $(\mathbf{x}, y) \sim P$ be an individual in P represented by a d-dimensional feature set and a binary label $y \in \{0, 1\}$. In the AFA setting, $\mathbf{x}^{(q)} \subset \mathbf{x}$ denotes a query on a subset of features in $\mathbf{x}$, with $q \subset \{0, ..., d\}$, and $\mathbf{x}^{(q)}$ the partial feature vector. The decision maker incurs a cost for the collected features $c^{(q)} = \sum_{j\in q} c_j$. The cost vector $\mathbf{c}$ represents the cost of each feature and is the same for each individual in P. It can represent different types of costs that the decision maker or an individual might incur when a feature is queried such as monetary and privacy costs.

We study the context in which a decision maker can choose what information to collect about an individual in order to maximize accuracy while ensuring fairness. Across all individuals in P, the decision maker is constrained by an average information budget:

Definition 1. *The information budget $\bar{b}$ is a global constraint that represents the average budget that can be used across individuals in P, $\bar{b} = \frac{1}{n}\sum_{i\in P} b_i$ with $b_i = \sum_{j\in q} c_j$, the information budget used for feature collection for a single individual i in P.*

In our population P we have a set of k disjoint population subgroups $G_1, \ldots, G_k$ defined over the protected attributes (such as certain combinations of protected attribute values like race and gender) across which we measure fairness. Note that the number of population subgroups is exponential in the number of protected attributes (e.g. three binary protected attributes will lead to $2^3 = 8$ subgroups). Generally, these subgroups will have different base rates μ_t, which represents the probability of belonging to the positive class $\mu_t = P_{(\mathbf{x},y)\sim G_t}[y = 1]$ across individuals in group t. For classification, we have a separate probabilistic classifier for each group G_t, $h_t : \mathbb{R}^k \rightarrow [0, 1]$. In practice, these separate classifiers are stemming from a single classifier trained on P and only differ because of subgroup-specific calibration. For the probabilistic error rates as well as for measuring disparity, we follow the generalized definitions introduced in [25]:

Definition 2. *The generalized false-positive rate for classifier h_t is $c_{fp}(h_t) = \mathbb{E}_{(\mathbf{x},y)\sim G_t}[h_t(\mathbf{x}^{(q)}) \mid y = 0]$. The generalized false-negative rate is $c_{fn}(h_t) = \mathbb{E}_{(\mathbf{x},y)\sim G_t}[1 - h_t(\mathbf{x}^{(q)}) \mid y = 1]$.*

If the classifier would output binary predictions instead of probabilities, these rates would simply represent standard false-positive and false-negative rates. Similarly, we use generalized notions of equalized odds and equal opportunity for probabilistic classifiers:

Definition 3. *Equal opportunity for a set probabilistic classifiers $h_1, \ldots, h_k$ for groups $G_1, \ldots, G_k$ requires $c_{fn}(h_t) = c_{fn}(h_{t'})$ for all possible combinations of t and t'. Equal odds requires both $c_{fn}(h_t) = c_{fn}(h_{t'})$ and $c_{fp}(h_t) = c_{fp}(h_{t'})$.*

For probabilistic classifiers, however, these two conditions do not ensure fairness if the classifier probabilities the classifier outputs are not calibrated. This is confirmed both theoretically and experimentally in [5, 6, 25].

Definition 4. *A classifier h_t is calibrated if $P_{(\mathbf{x},y)\sim G_t}[y = 1 \mid h_t(\mathbf{x}^{(q)}) = p] = p$.*

In Figure 1, we observe the set of calibrated classifiers for two groups G_1 and G_2. For each group, the classifiers lie on a line with slope $(1 - \mu_t)/\mu_t$ that connects the perfect classifier at the origin with the base rate classifier on the $c_{fp} + c_{fn} = 1$ line. The perfect classifier always assigns the correct prediction, while the base rate classifier has no predictive power and naively assigns the base rate to each individual [16, 25]. For an AFA classifier, the base rate classifier is simply the classifier before any features have been acquired $h(\mathbf{x}^{q=\emptyset})$.

4 EQUAL OPPORTUNITY

We will now derive a set of stopping criteria for each population subgroup that ensure satisfying equal opportunity. Intuitively, the stopping criteria should be chosen such that we collect more features for subgroups for which the model is less certain. By stopping later, we acquire more features, have more predictive power, and move down the slope in Figure 1 towards the perfect classifier at the origin. First, we reformulate c_{fn} from Definition 2 as

$$c_{fn}(h_t) = \frac{1}{\sum_{(\mathbf{x},y)\in G_t} \mathbb{1}_{y=1}} \sum_{(\mathbf{x},y)\in G_t} \mathbb{1}_{y=1}(1 - h_t(\mathbf{x}^{(\mathbf{q})})) \qquad (2)$$

The normalization can simply be replaced by a constant $1/(|G_t|\mu_t)$ since we marginalize over all $\mathbf{x}$ in G_t. Because we do not have access to ground truth labels $\mathbb{1}_{y=1}$ at test time, we replace them with the estimates from the the probabilistic classifier $h_t(x^{(q)})$:

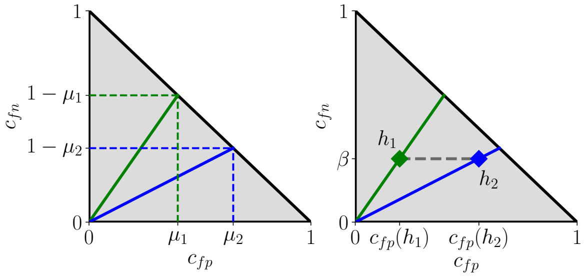


Figure 1: Left, we observe the set of calibrated classifiers h_1 and h_2 for G_1 in green and G_2 in blue. The base rates are $\mu_1 = 0.4$ and $\mu_2 = 0.65$. Right, we observe two classifiers h_1 and h_2 that satisfy calibration and equal opportunity with a target generalized false-negative rate β.

$$c_{fn}(h_t) = \frac{1}{|G_t|\mu_t} \sum_{(\mathbf{x},y)\in G_t} h_t(\mathbf{x}^{(q)})(1 - h_t(\mathbf{x}^{(\mathbf{q})})). \qquad (3)$$

One way to satisfy equal opportunity is to ensure that, in expectation, we have the same generalized false-negative rate c_{fn} for each group G_t such that $\mathbb{E}_{(\mathbf{x},y)\sim G_t}[c_{fn}(\mathbf{x}^{(q)})] = \beta \ \forall t$, where β can be chosen according to the information budget $\bar{b}$. To achieve this, we slowly increase the confidence of our classifier ($h_t(\mathbf{x}^{(q)}) \rightarrow 1$ or $h_t(\mathbf{x}^{(q)}) \rightarrow 0$) by sequentially adding features one-by-one. We stop collecting features when our probabilistic classifier crosses an upper or lower threshold probability, $h_t(\mathbf{x}^{(q)}) \geq \alpha_u$ or $h_t(\mathbf{x}^{(q)}) \leq \alpha_l$. For a desired β we can find these stopping thresholds α_u and α_l by ensuring equal $h_t(\mathbf{x}^{(q)})(1 - h_t(\mathbf{x}^{(\mathbf{q})})/\mu_t = \beta$ for every individual in our group G_t. Bringing everything except the classifier to one side of the equation, we want the probabilities to be $h_t(\mathbf{x}^{(q)}) = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - 4\beta\mu_t}$ which leads to thresholds

$$\alpha_u = \frac{1}{2} + \frac{1}{2}\sqrt{1 - 4\beta\mu_t}, \qquad \alpha_l = \frac{1}{2} - \frac{1}{2}\sqrt{1 - 4\beta\mu_t} \qquad (4)$$

Thus, by choosing the right stopping criteria for each individual $\mathbf{x}$ according to their subgroup-specific base rate μ_t, we ensure that we satisfy equal opportunity. See Figure 1 for an example with two subgroups. In practice, a decision maker would not choose the target rate β but, instead, tune β to meet an information budget $\bar{b}$. A higher information budget $\bar{b}$ allows for a lower target rate β.

Analogously, if we instead want to achieve equalized false-positive rates (predictive equality) across groups, we can derive a similar but different set of thresholds $\alpha_u = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 4\beta(\mu_t - 1)}$ and $\alpha_l = \frac{1}{2} - \frac{1}{2}\sqrt{1 + 4\beta(\mu_t - 1)}$. Finally, to achieve equal odds, we would have to find the same set of thresholds for both equal false-positive rates and equal false-negative rates. The only case for which these thresholds are the same is for $1 - \mu_t = \mu_t$ (i.e. $\mu_t = 0.5$) which is the trivial case for which there was already no unfairness. This confirms the conclusion in [25] that for different base rates, one cannot simultaneously achieve equal odds and calibration.

Table 1: Overview of the datasets and subgroups split by the protected attributes. Accuracy and AUC are computed on a dataset-level using the full feature set, while μ is the dataset-level base-rate $P(y)$. For each subgroup we compute the relative number of individuals n_t and the base rate μ_t.

Dataset						Subgroup$_1$			Subgroup$_0$		
Name	N$_{samples}$	N$_{feat}$	Acc	AUC	μ	Label$_1$	n$_1$	μ_1	Label$_0$	n$_0$	μ_0
Synthetic [9, 23]	10,000	150	85.9%	0.933	50.0%	$z = 1$	50.0%	41.3%	$z = 0$	50.0%	58.9%
Mexican poverty [12, 22]	70,305	182	78.7%	0.856	35.5%	Urban	63.6%	34.9%	Rural	36.4%	36.6%
Adult income [18]	49,000	14	86.3%	0.911	23.9%	White	85.4%	25.4%	Non-white	14.6%	15.3%

Assumptions. In this framework, we make two key assumptions. First, we assume that for each individual we have sufficient statistical power to reach the target β by simply adding more features. In practice, however, there will be a non-zero Bayes-optimal error rate such that we cannot reach the perfect classifier with $\beta = 0$ even with unlimited budget for feature acquisition. Second, we assume that the probabilities are exactly $p = \alpha_u$ or $p = \alpha_l$ while in reality we stop when we cross the threshold and thus $p \geq \alpha_u$ or $p \leq \alpha_l$. In the experiments in Section 5 we show that relaxing both assumptions does not limit the effectiveness of our framework.

4.1 Implications

This result is important for several reasons. First, it provides a theoretical framework for understanding the results presented in [22]. In the *active fairness* framework described there, optimization is used to find a set of parameters that allows for equal opportunity and calibration in the AFA setting, but lacks a theoretical underpinning.

Second, we only need a subgroup's base rate to find α_u and α_l. This is crucial when the problem is extended to a case with several multi-class protected attributes, like gender, race, and sexual orientation. If one instead would try to find the parameters by optimizing over a budget and fairness constraints for each protected attribute, the resulting classifier could contain intra-group unfairness.

Third, comparing this result to the randomization approach presented in [25], our framework shows that by using AFA, we can achieve fairness in a budget-constrained setting without having to resort to randomized approaches that are inefficient, pareto-suboptimal, and lead to intra-group unfairness.

5 EXPERIMENTS

In light of these findings, we demonstrate the effectiveness and limitations of our framework on one synthetic and two public real-world datasets. In this section we aim to satisfy equal opportunity (equal false-negative rates) but in Appendix A we demonstrate that the method can also be used for satisfying predictive equality (equal false-positive rates).

5.1 Implementation

Implementation requires two elements, a probabilistic model and a feature acquisition strategy.

Probabilistic model. First, we need a model that allows us to estimate $P(y|\mathbf{x}^{(q)})$ for arbitrary feature subsets $\mathbf{x}^{(q)}$, with $q \in [0, d]$. We implement this using a probabilistic random forest, designed to deal with incomplete data in trees [26]. Specifically, we first train a standard random forest using the complete feature vector $\mathbf{x}$ for each individual in our training set. At test time, however, we now only have access to part of the feature vector $\mathbf{x}^{(q)}$. In a probabilistic random forest, when the algorithm encounters a tree node for which the value is missing in the feature vector $\mathbf{x}^{(q)}$, the algorithm continues along both branches towards the leafs while the outcomes in each branch are weighted based on the estimated probability for the missing value. For each individual, that probability is estimated from the frequency of values in the training set. We then compute classification probabilities as a weighted average of the leaf purity across all leaves landed on by the search. Finally, the predicted probability is averaged across all trees. Analogously, gradient boosting and other models can be adjusted to admit incomplete feature vectors [26, 31]. In this work, all random forests are created using `scikit-learn` with 64 trees and maximally 150 leaf nodes. Additionally, we built a custom predict function that works with the `scikit-learn` object but accounts for the missing feature values.

Feature acquisition strategy. Second, we implement an efficient feature acquisition strategy to estimate which next feature can be best selected based on the current partially observed feature vector $\mathbf{x}^{(q)}$, while balancing cost and increasing accuracy. We implement a *greedy* feature selection algorithm based on the expected utility methods described in [13, 17]. For an individual with feature vector $\mathbf{x}$, and at each feature collection iteration, the algorithm searches for the feature $j' \notin q$ that maximizes the difference between the current predicted probability $\hat{P}$ and the expected probability given that an additional feature j' is queried with cost c_j, given by:

$$j' = \underset{\{j : j \notin q, j \in [0,d]\}}{\arg\max} \frac{1}{c_j} \left| \hat{P}\{y = 1|\mathbf{x}^{(q \cup j)}\} - \hat{P}\{y = 1|\mathbf{x}^{(q)}\} \right|. \quad (5)$$

5.2 Datasets

An overview of the datasets is given in Table 1. All results are computed using random 60%/20%/20% train/validation/test splits. The Synthetic dataset is generated using the `make_classification` function from `scikit-learn` [9, 23] where we use the default set of parameters while setting `class_sep` to 1.5 (default is 1.0) to make the task slightly easier. The protected attribute is a randomly selected feature which we exclude from the dataset and binarize by splitting along the median. The Mexican Poverty dataset is extracted from the 2016 publicly available Mexican household survey containing household binary poverty levels for prediction, as well as a series of household features [12]. We will release the processed dataset. Finally, we use the Adult Income dataset from UCI Machine Learning Repository [18] which comprises demographic and occupational attributes, with the goal of classifying whether a person's income is above $50,000.

5.3 Achieving equal opportunity

We empirically demonstrate that our framework satisfies equal opportunity for a given information budget $\bar{b}$. To make the results more interpretable, we choose the costs to be the same for each feature $c_j = 1$. Hence, the budget $\bar{b}$ is simply the average number of features that can be queried across individuals. We also tested the framework with linearly increasing feature costs and feature costs drawn from a normal distribution, while observing similar behavior. To ensure calibrated probabilities, we fit a sigmoid function to the classifier's scores using the validation set; a calibration method known as Platt scaling [24].

Figure 2 demonstrates that we can satisfy equal opportunity for the three datasets. In Figure 2a we plot the derived equal opportunity classifiers in the generalized false-positive/false-negative plane with 5%, 10%, and 20% as information budgets for respectively the Synthetic, Adult Income, and Mexican Poverty datasets. Table 2 shows the residual false-negative disparities after applying our stopping criteria as well as the false-positive disparity. The results in the table are benchmarked against the disparities between groups when the classifiers have access to all features (i.e. no stopping criteria). As expected, our framework leads to drastically lower false-negative disparities while the false-positive disparities are similar to the baseline. In Table A1 we find the stopping criteria, budgets, and classifier performance (using area under the ROC curve) for equal opportunity, predictive equality and the baseline model using all features.

Figure 2b demonstrates that our framework allows for achieving equal opportunity for a range of different information budgets. The steepest decrease in c_{fn} is observed for smaller information budgets because our feature acquisition strategy chooses the most predictive features first. For larger budgets, the curves plateau as the additional features do not further increases the classifier performance.

Our framework assumes a one-to-one mapping between the target β and the actual generalized-false negative rate c_{fn}. For the Synthetic dataset in Figure 2c, we indeed observe an approximate one-to-one mapping between target and actual. For the Mexican Poverty dataset, however, we observe a strong positive correlation but the actual false-negative rate increases slower than the target rate. For small target rates β this is the result of lower overall classification performance for the Mexican dataset (AUC 0.85 when using all features versus 0.93 for the Synthetic dataset); as β becomes smaller, the thresholds α_u and α_l approach 1 and 0. Therefore, when the classification performance is low, many instances will fail to meet the stopping criteria before running out of possible features to query which increases the actual rate c_{fn}. For high values of β, another effect is at play. The smaller than expected c_{fn} is observed because our probabilities do not end up exactly at α_u and α_l (our stopping criteria are defined as $h_t(\mathbf{x}^{(\mathbf{q})}) \geq \alpha_u$, not as $h_t(\mathbf{x}^{(\mathbf{q})}) = \alpha_u$). Importantly, however, we observe that these assumptions do not affect our ability to achieve equal opportunity.

Table 2: Information budgets $\bar{b}$ and absolute differences (disparities) in generalized false-negative $|\Delta c_{fn}|$ and false-positive rates $|\Delta c_{fp}|$ for the equal opportunity classifiers visualized in Figure 2. We benchmark our framework to the classifiers with access to all features x ($\bar{b}$ = 100%).

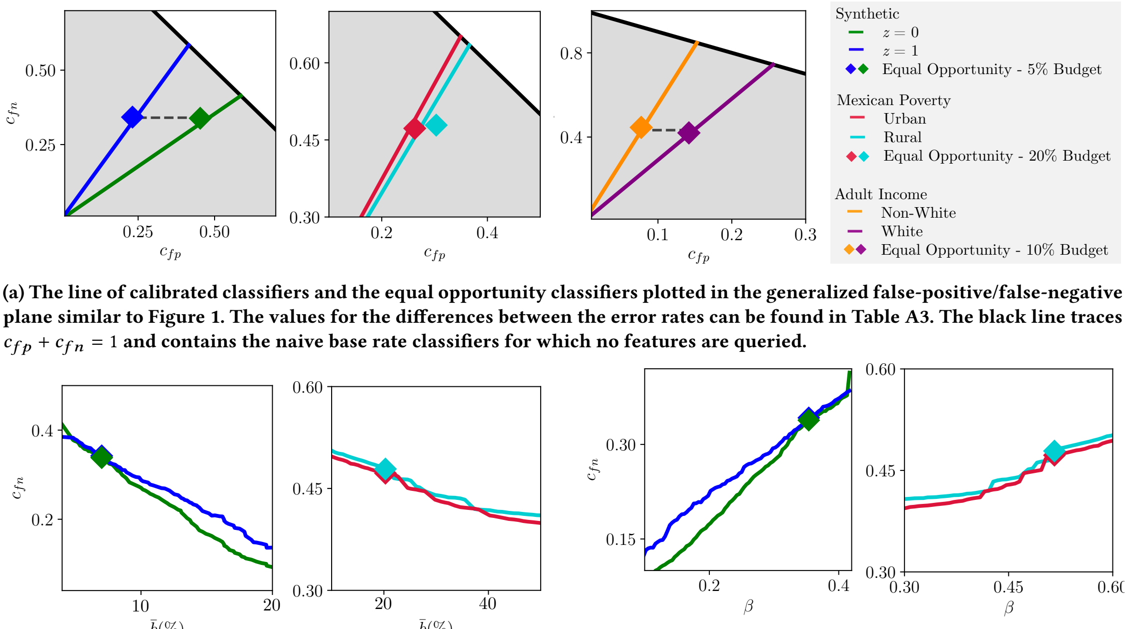
	Equal opportunity			All features		
Dataset	$\bar{b}$	$\lvert\Delta c_{fn}\rvert$	$\lvert\Delta c_{fp}\rvert$	$\bar{b}$	$\lvert\Delta c_{fn}\rvert$	$\lvert\Delta c_{fp}\rvert$
Synthetic	5%	0.0039	0.221	100%	0.097	0.042
Mexican Pov.	20%	0.0063	0.040	100%	0.019	0.042
Adult income	10%	0.026	0.065	100%	0.038	0.056

Finally, we test our framework for eight disjoint subgroups defined over three protected household attributes (Young/Old, Urban/Rural, and With/Without Children) in the Mexican Poverty dataset. When using optimization for achieving fairness, large intragroup unfairness can manifest itself; even though disparities measured across protected attributes are small, large differences between false-negative rates for each subgroup defined over the attributes can exist [14]. In contrast, our framework requires all eight false-negative rates to be approximate equal and, indeed, empirically we observe that all fall within the [0.440, 0.541] range. See Tables A2 and A3 in the appendix for an overview of results for the three protected attributes.

6 CONCLUSION

We introduced a framework for achieving equal opportunity (and predictive equality) for calibrated probabilistic classifiers in an active feature-value acquisition setting. The framework relates a target generalized false-negative rate and a subgroup-specific base rate to a set of stopping criteria, used to determine when to stop querying additional features for fair classification. The target false-negative rate can be tuned using the available information budget. The relationship between error and base rates is intuitive as base rate differences are what give rise to disparities between calibrated classifiers. On three datasets, we show the effectiveness of the framework and demonstrate that relaxing some of the assumptions in our framework does not significantly change its effectiveness.

Importantly, the proposed framework neither relies on optimization nor any form of randomization. Furthermore, it is not susceptible to intra-group unfairness and provides a new perspective on how we could combine individual and statistical notions of fairness. The ability to set the expected false-negative rates for each subgroup simply by deriving a set of stopping criteria could be used to ensure statistical notions of fairness to hold not only for a small number of larger subgroups but potentially for an exponential number of smaller subgroups. This could enable a set of classifiers for which both individual and statistical notions of fairness hold without having to collect the protected attributes. In turn, this allows for fair decision making in contexts where one deals with a multitude of subgroups or when collecting the protected attributes is unethical or impossible.



(a) The line of calibrated classifiers and the equal opportunity classifiers plotted in the generalized false-positive/false-negative plane similar to Figure 1. The values for the differences between the error rates can be found in Table A3. The black line traces $c_{fp} + c_{fn} = 1$ and contains the naive base rate classifiers for which no features are queried.

(b) Generalized false-negative rates c_{fn} for equal opportunity classifiers along a range of different information budgets $\bar{b}$.

(c) Generalized false-negative rates c_{fn} for different target false-negative rates β. Ideally, you expect a straight-line relationship with slope 1.

Figure 2: For the datasets described in Table 1, we demonstrate equal opportunity for three different budgets. For each subgroup, we show the possible set of calibrated classifiers (lines) together with the specific classifier that achieves equal opportunity for the given budget (diamonds).

REFERENCES

[1] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2019. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. *arXiv* (2019).

[2] Flavio P. Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. 2017. Optimized Pre-Processing for Discrimination Prevention. In *Advances in Neural Information Processing Systems*. 3992–4001.

[3] Aaron Chalfin, Oren Danieli, Andrew Hillis, Zubin Jelveh, Michael Luca, Jens Ludwig, and Sendhil Mullainathan. 2016. Productivity and selection of human capital with machine learning. *American Economic Review* 106, 5 (2016).

[4] Irene Chen, Fredrik D Johansson, and David Sontag. 2018. Why Is My Classifier Discriminatory?. In *Advances in Neural Information Processing Systems*. 3539.

[5] Sam Corbett-Davies and Sharad Goel. 2018. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *arXiv* (2018).

[6] Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau. 2016. Assessing calibration of prognostic risk scores. *Statistical methods in medical research* 25, 4 (2016), 1692–1706.

[7] Cynthia Dwork, 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, 214–226.

[8] Tianshi Gao and Daphne Koller. 2011. Active classification based on value of classifier. In *Advances in Neural Information Processing Systems*.

[9] Isabelle Guyon. 2003. Design of experiments of the NIPS 2003 variable selection benchmark.

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

[11] Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. 2007. Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications* 33, 4 (2007), 847–856.

[12] Pablo Ibarrarán, Nadin Medellín, Ferdinando Regalia, Marco Stampini, Sandro Parodi, Luis Tejerina, Pedro Cueva, and Madiery Vásquez. 2017. How Conditional Cash Transfers Work. (2017).

[13] Pallika Kanani and Prem Melville. 2008. Prediction-time active feature-value acquisition for cost-effective customer targeting. *Advances In Neural Information Processing Systems (NIPS)* (2008).

[14] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2017. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv* (2017).

[15] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. 2015. Prediction policy problems. *American Economic Review* 105, 5 (2015).

[16] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint* (2016).

[17] Balaji Krishnapuram, Shipeng Yu, and R Bharat Rao. 2011. *Cost-sensitive Machine Learning*. CRC Press.

[18] Moshe Lichman et al. 2013. UCI machine learning repository.

[19] Li-Ping Liu, Yang Yu, Yuan Jiang, and Zhi-Hua Zhou. 2008. TEFE: A time-efficient approach to feature extraction. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE.

[20] Pranay K. Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R. Varshney, and Ruchir Puri. 2019. Bias Mitigation Post-Processing for Individual and Group Fairness. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2847–2851.

[21] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2015. The variational fair autoencoder. *arXiv* (2015).

[22] Alejandro Noriega-Campero, Michiel Bakker, Bernardo Garcia-Bulle, and Alex Pentland. 2019. Active Fairness in Algorithmic Decision Making. *Proceedings of AAAI / ACM Conference on Artificial Intelligence, Ethics, and Society* (2019).

[23] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[24] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*

10, 3 (1999), 61–74.

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

[26] Maytal Saar-Tsechansky and Foster Provost. 2007. Handling missing values when applying classification models. *Journal of machine learning research* 8, Jul (2007), 1623–1657.

[27] Prasanna Sattigeri, Samuel C. Hoffman, Vijil Chenthamarakshan, and Kush R. Varshney. 2019. Fairness GAN: Generating Datasets with Fairness Properties using a Generative Adversarial Network. In *ICLR Workshop on Safe Machine Learning*.

[28] Hajin Shim, Sung Ju Hwang, and Eunho Yang. 2018. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*. 1368–1378.

[29] Eric Siegel. 2013. *Predictive analytics: The power to predict who will click, buy, lie, or die.* Wiley Hoboken.

[30] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 1–7.

[31] David Williams, Xuejun Liao, Ya Xue, and Lawrence Carin. 2005. Incomplete-data classification using logistic regression. In *Proceedings of the 22nd International Conference on Machine learning*. 972–979.

[32] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. *arXiv* (2017).

[33] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. *arXiv preprint* (2017).

A ACHIEVING PREDICTIVE EQUALITY

In line with satisfying equal opportunity in the main text, we empirically demonstrate that our framework satisfies predictive equality (equal false-positive rates) for three different information budgets 10%, 15%, and 30% for respectively the Synthetic, Adult Income, and Mexican Poverty datasets. In Table A1 we observe the statistics for both equal opportunity and predictive equality. In agreement with equal opportunity, we see a drastic decrease in target error rate (now $|\Delta c_{fp}|$) with respect to the false-positive disparity measured across the benchmark classifiers that have access to all features.

Table A1: Comparison of AUC, absolute differences in generalized false-negative $|\Delta c_{fn}|$ and false-positive $|\Delta c_{fp}|$ rates across the equal opportunity, predictive equality and benchmark classifiers for the three different datasets. The equal opportunity and predictive equality classifier were derived by setting a group-specific threshold and applying active feature acquisition while the benchmark classifier has access to the complete feature set. The upper threshold α_u is shown while the lower threshold relates to the upper threshold as $\alpha_l = 1 - \alpha_u$. Both are determined by the average information budget $\bar{b}$.

	Equal opportunity						Predictive equality						All features			
Dataset	$\bar{b}$	$\|\Delta c_{fn}\|$	$\|\Delta c_{fp}\|$	AUC	$\alpha_{u,1}$	$\alpha_{u,0}$	$\bar{b}$	$\|\Delta c_{fn}\|$	$\|\Delta c_{fp}\|$	AUC	$\alpha_{u,1}$	$\alpha_{u,0}$	$\bar{b}$	$\|\Delta c_{fn}\|$	$\|\Delta c_{fp}\|$	AUC
Synthetic	5%	0.0039	0.221	0.77	0.82	0.71	10%	0.225	0.002	0.77	0.69	0.81	100%	0.097	0.042	0.933
Mexican Poverty	20%	0.0063	0.040	0.78	0.77	0.75	30%	0.038	0.011	0.79	078	0.79	100%	0.019	0.042	0.856
Adult Income	10%	0.026	0.065	0.86	0.78	0.89	15%	0.423	0.010	0.81	0.78	0.73	100%	0.038	0.056	0.911

Table A2: Active feature acquisition for eight different subgroups defined over three binary protected attributes in the Mexican Poverty dataset. The metrics c_{fn}, c_{fp} and AUC are computed on each subgroup level with a 25% information budget $\bar{b}$. Each subgroup has its own threshold as stopping criterion based on the subgroup specific base rate μ_t. Furthermore, we report the relative number of individuals n_t with respect to the whole set and the fairness statistics for the benchmark case.

			Equal opportunity			All features		
Subgroup	n_t	μ_t	c_{fn}	c_{fp}	AUC	c_{fn}	c_{fp}	AUC
Young ∩ Urban ∩ With Children	20.0 %	51.4%	0.465	0.460	0.667	0.305	0.306	0.848
Young ∩ Urban ∩ Without Children	13.4%	21.6%	0.502	0.174	0.828	0.494	0.140	0.866
Young ∩ Rural ∩ With Children	13.5%	50.3%	0.443	0.446	0.699	0.333	0.349	0.812
Young ∩ Rural ∩ Without Children	5.8%	23.0%	0.541	0.186	0.773	0.559	0.153	0.810
Old ∩ Urban ∩ With Children	7.4%	54.0%	0.448	0.468	0.681	0.320	0.307	0.750
Old ∩ Urban ∩ Without Children	22.4%	21.7%	0.543	0.188	0.810	0.542	0.160	0.838
Old ∩ Rural ∩ With Children	4.5%	49.5%	0.440	0.433	0.711	0.339	0.322	0.817
Old ∩ Rural ∩ Without Children	12.7%	24.1%	0.530	0.224	0.785	0.531	0.191	0.804
$\cup_{Subgroups}$	100%	35.5%	0.480	0.283	0.794	0.432	0.233	0.824

Table A3: Absolute differences in generalized false-negative $|\Delta c_{fn}|$ and false-positive $|\Delta c_{fp}|$ rates on a group-level. The thresholds for feature acquisition were set on a subgroup level (same as in Table 4). Controlling for error rates (in this case c_{fn}) on a subgroup level leads to fairness on the level of the sensitve attribute.

Group	$\|\Delta c_{fn}\|$	$\|\Delta c_{fp}\|$
Young/Old	0.045	0.088
Urban/Rural	0.070	0.097
With/Without Children	0.089	0.257

Chapter 4

DADI: Dynamic Discovery of Fair Information with Adversarial Reinforcement Learning

Contributing Article: Bakker, A. M., Tu, D. P., Riveron Valdes, H., Krishna, G., Varshney, K., Weller, A., Pentland, A. S. DADI: Dynamic Discovery of Fair Information with Adversarial Reinforcement Learning. *Human-Centric Machine Learning Workshop at NeurIPS 2019.*

Declaration of Contributions

	Bakker, Michiel	**Tu, Duy Patrick**	Riveron Valdes, Humberto	Gummadi, Krishna	Varshney, Kush	Weller, Adrian	Pentland, Alex "Sandy"
Development of idea	x						
Development of theory	x						
Provided helpful discussions and input	x	**x**	x	x	x	x	
Implementation of framework		**x**					
Data processing and experiments		**x**					
Analysis of results		**x**					
Writing of the manuscript	x	**x**					
Improvement of text	x	**x**	x		x	x	
Supervision of research					x	x	x
Presentation of research	x	**x**					

DADI: Dynamic Discovery of Fair Information with Adversarial Reinforcement Learning

Michiel A. Bakker
Massachusetts Institute of Technology
MIT-IBM Watson AI Lab
bakker@mit.edu

Duy Patrick Tu[*]
Massachusetts Institute of Technology
Ludwig-Maximilians-Universität München
patrick2@mit.edu

Humberto Riverón Valdés
Massachusetts Institute of Technology
MIT-IBM Watson AI Lab
hriveron@mit.edu

Krishna P. Gummadi
Max Planck Institute for
Software Systems
gummadi@mpi-sws.org

Kush R. Varshney
IBM Research
MIT-IBM Watson AI Lab
krvarshn@us.ibm.com

Adrian Weller
University of Cambridge
Alan Turing Institute
aw665@cam.ac.uk

Alex 'Sandy' Pentland
Massachusetts Institute of Technology
MIT-IBM Watson AI Lab
pentland@mit.edu

Abstract

We introduce a framework for dynamic adversarial discovery of information (DADI), motivated by a scenario where information (a feature set) is used by third parties with unknown objectives. We train a reinforcement learning agent to sequentially acquire a subset of the information while balancing accuracy and fairness of predictors downstream. Based on the set of already acquired features, the agent decides dynamically to either collect more information from the set of available features or to stop and predict using the information that is currently available. Building on previous work exploring adversarial representation learning, we attain group fairness (demographic parity) by rewarding the agent with the adversary's loss, computed over the final feature set. Importantly, however, the framework provides a more general starting point for fair or private dynamic information discovery. Finally, we demonstrate empirically, using two real-world datasets, that we can trade-off fairness and predictive performance.

1 Introduction

There are two parties involved in information transfer: a *data owner* who has ownership over its own data or data it holds on behalf of others and a *data collector* who is tasked with collecting the most informative set of data, often to maximize the performance of some predictor downstream. Intentionally or otherwise, this process of data collection and prediction can lead to biases that unfairly favor one protected subgroup over another. Numerous recent studies have shown that naively optimizing for predictive performance can lead to unfair prediction outcomes in high-stake domains such as criminal justice, credit assessment, recruiting, and healthcare [Kleinberg et al., 2017, Chalfin et al., 2016, Huang et al., 2007, Obermeyer et al., 2019].

[*] Duy Patrick Tu did this work while visiting MIT from LMU Munich.

Workshop on Human-Centric Machine Learning at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada. (non-archival)

Consequently, the data owner faces a critical decision: if it cannot trust the data collector, which information should it share to ensure fair decision making? While the optimal strategy to maximize predictive performance is to naively share all the data available, the data owner has to be more careful when it wants to ensure that the predictions downstream are fair. Removing the sensitive attribute is the most obvious strategy, but is ineffective when the attribute is redundantly encoded in other features [Dwork et al., 2012]. Another strategy is to first apply *fair feature selection* in which one formulates an optimization problem to select a subset of features that maximizes accuracy, given a maximum unfairness constraint [Grgić-Hlača et al., 2018]. This strategy, though effective, is inefficient as it removes each feature simultaneously for all individuals, ignoring any differences in the underlying conditional dependencies. For example, for individuals that live in Chicago, the most racially segregated city in America, zipcode will be highly correlated with race and using this feature can thus lead to racially biased predictions [Logan, 2014]. In contrast, if an individual lives in Irvine, California, America's most racially integrated city, zipcode alone will not reveal an individual's race. Removing zipcode for all individuals is therefore an effective but inefficient strategy to ensure fairness.

Motivated by this problem, we propose the DADI (Dynamic Adversarial Discovery of Information) framework as a general sequential information acquisition framework for any task. Our contributions are as follows: to the best of our knowledge, we introduce the first framework for dynamic adversarial discovery of information which we utilize to acquire feature sets that ensure fair decision making. In this framework, we formulate the feature acquisition task as a minimax optimization problem in which a reinforcement learning (RL) agent simultaneously minimizes the classification loss while maximizing the loss of an adversary. We actualize this with a joint framework that simultaneously trains a classifier, an adversary, and an RL agent using deep Q-learning. Building on work on adversarial representation learning, we investigate the effects of two different adversarial reward functions to achieve *demographic parity* [Edwards and Storkey, 2016, Madras et al., 2018]. Finally, we demonstrate the effectiveness of our framework with two real-world public datasets.

2 Related Work

Fairness Recent years have seen an explosion in academic work that seeks to define and obtain fairness in automated decision making systems. At a high level, this literature has focused on two families of definitions: *statistical* notions of fairness and *individual* notions of fairness [Dwork et al., 2012, Verma and Rubin, 2018]. Most of the literature, including this work, focuses on statistical or group definitions of fairness, in which we require parity of some statistical measure to hold across a small number of protected subgroups. In contrast, individual fairness definitions have no notion of protected subgroups, but instead formulate constraints that bind on pairs of individuals [Dwork et al., 2012, Joseph et al., 2016]. Both families of definitions have strengths and weaknesses; statistical notions are easy to verify but do not provide any guarantees to individuals, while individual notions do give individual guarantees but are difficult to implement in practice and are ambiguous with respect to the agreed-upon distance function.

In this work, we focus on *demographic parity*, requiring parity of the positive classification rate across groups, i.e. $P(\hat{y} = 1 \mid b = 0) = P(\hat{y} = 1 \mid b = 1)$, where $\hat{y} \in \{0, 1\}$ is the binary prediction of a model that classifies feature set $\mathbf{x}$ and $b \in \{0, 1\}$ is the sensitive attribute. The usefulness of demographic parity is limited when the base rate differs across groups, i.e. $P(y = 1 \mid b = 0) \neq (y = 1 \mid b = 1)$ where $y \in \{0, 1\}$ is the ground truth label. In that case, the metric can be generalized by conditioning on the ground truth label, yielding equal false negative rates (*equal opportunity*) or equal false negative and false positive rates (*equal odds*) as measures of fairness [Hardt et al., 2016]. We demonstrate the effectiveness of our framework using demographic parity, but note that alternative adversarial objectives have been introduced that can be combined with our framework to achieve equal opportunity or equal odds [Madras et al., 2018].

Adversarial training Adversarial training for deep generative models was introduced in Goodfellow et al. [2014], framing the learning as a two-player game between a generator and a discriminator. The generator aims to fool the discriminator by generating fake data that resembles data from a dataset X while the discriminator is trained to distinguish between 'real' data from and 'fake' data generated by the generator. Learning proceeds using a minimax optimization where the generator and discriminator are optimized jointly. At each iteration, the discriminator improves its ability to discriminate between real and fake which, in turn, forces the generator to generate fake data that better resembles the real data.

Adversarial training was first applied in the context of fairness by Edwards and Storkey [2016], proposing adversarial training to ensure that multiple distinct data distributions from different demographic subgroups are modeled as a single representation. The discriminator aims to distinguish between subgroups while an encoder aims to map each data distribution to a single representation to fool the discriminator. Subsequently, these representations can be safely shared with a data collector while ensuring demographic parity for predictions downstream. Beutel et al. [2017] further explores this approach in the context of demographically imbalanced data. Finally, Zhang et al. [2018] and Madras et al. [2018] extend this body of work by connecting multiple statistical notions of fairness to different adversarial objectives. Whereas the method presented in Zhang et al. [2018] predicts the sensitive attribute from the prediction of the classifier, our work is closer to the method in Madras et al. [2018], working directly with the learned representation. This allows for transferable representations that ensure fair outcomes for other third-party classifiers downstream.

Although this work is similar in spirit to adversarial representation learning, we aim to dynamically collect a fair subset of features instead of learning to map the full feature set to a fair representation. The ability to collect raw features instead of mapping to a representation is crucial for integration with current information systems where the collected information is used or audited by both human and machine decision makers downstream. If we consider our the example of credit assessment, a bank not only wants to collect a low-level abstract representation for the purpose of the initial creditworthiness prediction but also wants to explain the credit decisions to an applicant and store the applicant's information in a database to provide other services downstream or allow for audits.

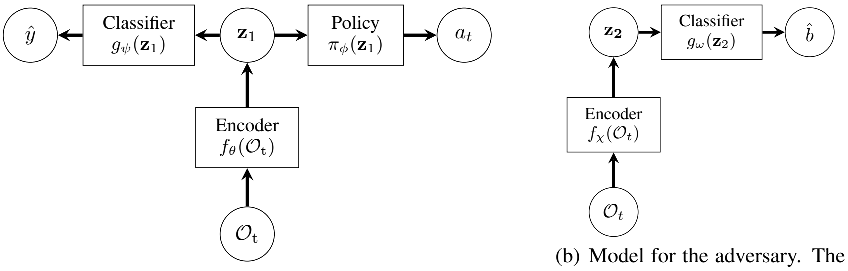
Active feature-value acquisition Different from *active learning*, active feature-value acquisition (AFA) is concerned with feature-wise active learning for each instance. AFA is of great need in cost-sensitive applications where the data collector needs to balance an available information budget with predictive accuracy. A traditional AFA system consists of three components: 1) a classifier that can handle partially observed feature sets, 2) a strategy for determining which feature to select next based on the features that are already collected, and 3) a stopping criterion for determining when to stop acquiring more features and make a final prediction.

First, there are different ways a classifier can handle a partial features set. Generative models handle missing features naturally by first integrating out the variables while in discriminate models feature imputation or expectation-maximization can be used to first replace the missing values with estimates. In this work, we use a set encoder based on Vinyals et al. [2015] to encode arbitrary subsets of features. Second, to determine which feature to select next, we need a method that estimates the value of each of the unselected features based on the features that we have already collected. A recent approach, Efficient Dynamic Discovery of High-Value Information (EDDI), uses a partial variational autoencoder to represent the set of already acquired features. It then computes the mutual information between the current representation and each of the available features to select the feature that minimizes this information [Ma et al., 2019]. Finally, a stopping criterion is not specified in EDDI and most other AFA methods. However, some prior work assumes a fixed feature budget per individual after which the process terminates [Krishnapuram et al., 2011]. The *active fairness* framework presented in Noriega-Campero et al. [2019] extends this to group-specific budgets that are found to attain equal opportunity (equal false-positive or false-negative rates).

To effectively trade-off fairness and accuracy, we need a unified framework that jointly optimizes both the acquisition strategy and the stopping criterion. We adopt the framework from Shim et al. [2018] and model the feature acquisition process as a Markov decision process (MDP) where the action space consists of the set of unselected set of features and an additional STOP action which, upon selection, terminates the acquisition process. To ensure fairness, we formulate a reward function that balances low classification loss with a high adversarial loss.

3 Adversarial Discovery of Fair Information

Problem setup The setup of our framework most follows the joint active feature acquisition and classification framework in [Shim et al., 2018]; however, we extend their framework for use with an adversary. Let $(\mathbf{x}^{(i)}, y^{(i)}, b^{(i)}) \sim P$ be individual i in P represented by a d-dimensional feature vector


$\hat{y}$
Classifier $g_\psi(\mathbf{z}_1)$
$\mathbf{z}_1$
Policy $\pi_\phi(\mathbf{z}_1)$
a_t
Encoder $f_\theta(\mathcal{O}_\mathrm{t})$
$\mathcal{O}_\mathrm{t}$
$\mathbf{z_2}$
Classifier $g_\omega(\mathbf{z}_2)$
$\hat{b}$
Encoder $f_\chi(\mathcal{O}_t)$
$\mathcal{O}_t$

(a) Joint model for the label classifier and policy. The encoder f_θ maps the set of observed features $\mathcal{O}_t$ to the latent representation $\mathbf{z}_1$. From this the classifier g_ψ predicts $\hat{y}$ while the policy π_ϕ predicts the action a_t (the next feature to select or STOP).

(b) Model for the adversary. The encoder f_χ maps the set of observed features $\mathcal{O}_t$ to the latent representation $\mathbf{z}_2$ from which the classifier g_ω predicts the sensitive attribute $\hat{b}$.

Figure 1: Joint framework for dynamic adversarial discovery of information (DADI)

$\mathbf{x}^{(i)} \subseteq \mathbb{R}^d$, a binary label $y^{(i)} \in \{0, 1\}$, and a binary sensitive attribute $b^{(i)} \in \{0, 1\}$. We acquire the features in sequential order starting with an empty set $\mathcal{O}_0 := \emptyset$ at time $t = 0$. At every later timestep t, we choose a subset of features from the unselected set of features, $\mathbf{S}_t^{(i)} \subseteq \{1, \ldots, d\} \setminus \mathcal{O}_{t-1}^{(i)}$. After each new acquisition step, the classifier will have access to feature values in $\mathcal{O}_t^{(i)} := \mathcal{S}_t^{(i)} \cup \mathcal{O}_{t-1}^{(i)}$. We keep acquiring features up to time $T^{(i)}$ when we meet a stopping criterion. At that point, we will classify $\mathbf{x}^{(i)}$ using only the set of features in $\mathcal{O}_{T^{(i)}}^{(i)}$. Note that the specific set of selected features $\mathcal{O}_{T^{(i)}}^{(i)}$ will generally be different for each individual i. To learn the model that minimizes classification loss while maximizing the loss of the adversary we formulate the following optimization problem.

$$\underset{\psi,\theta,\omega,\chi}{\text{minimize}} \frac{1}{|P|} \sum_{i \in P} (1-\gamma)\mathcal{L}_C \left(g_\psi(f_\theta(\mathcal{O}_T^{(i)}), y^{(i)} \right) - \gamma \mathcal{L}_A \left(g_\omega(f_\chi(\mathcal{O}_T^{(i)}), b^{(i)} \right) \tag{1}$$

Where $\mathcal{L}_C$ and $\mathcal{L}_A$ are the suitable losses for the label classifier and the adversary. The encoder f_θ feeds into a classifier g_ψ for the label prediction $\hat{y}$ while f_χ and g_ω are the encoder and classifier for the sensitive attribute prediction $\hat{b}$. Hyperparameter γ specifies the desired balance between classification performance and fairness. When clear from context, we drop the superscript (i).

Markov decision process We define a Markov decision process (MDP) to find the set of features $\mathcal{O}_T^{(i)}$ that minimizes the objective in Eq. 1. For each episode, the state at time t is represented by the set of selected features $\{x_j\}_{j \in \mathcal{O}_t}$. The size of the state space is 2^d, the powerset of the feature set. At each timestep t, the action space consists of the set of unselected features $\{1, \ldots, d\} \setminus \mathcal{O}_{t-1}$ and an additional STOP action which, upon selection, stops the acquisition process after which the rewards are computed. The agent's reward function, computed at end of the episode for individual i, corresponds to

$$r(\mathcal{O}_T^{(i)}) = -(1-\gamma)\mathcal{L}_C\big(g_\psi(f_\theta(\mathcal{O}_T^{(i)}), y^{(i)})\big) + \gamma \mathcal{L}_A\big(g_\omega(f_\chi(\mathcal{O}_T^{(i)}), b^{(i)})\big) \tag{2}$$

where the first reward encourages accurate classification and the second reward encourages low mutual information between the feature set and the sensitive attribute. Now, if we now consider a policy π_ϕ^*, parametrized by ϕ, that is optimal for this MDP, then π_ϕ^* is also the optimal solution to the objective in Eq. 1. We can proof this by maximizing the aggregated reward in Eq. 2 over the population P

$$\arg\max_{\phi} \frac{1}{|P|} \sum_{i \in P} -(1-\gamma)\mathcal{L}_C\big(g_\psi(f_\theta(\mathcal{O}_T^{(i)}), y^{(i)})\big) + \gamma \mathcal{L}_A\big(g_\omega(f_\chi(\mathcal{O}_T^{(i)}), b^{(i)})\big) \tag{3}$$

$$= \arg\min_{\phi} \frac{1}{|P|} \sum_{i \in P} (1-\gamma)\mathcal{L}_C\big(g_\psi(f_\theta(\mathcal{O}_T^{(i)}), y^{(i)})\big) - \gamma \mathcal{L}_A\big(g_\omega(f_\chi(\mathcal{O}_T^{(i)}), b^{(i)})\big) \tag{4}$$

which is equivalent to the minimization objective in Eq. 1.

Generalized framework The generalized framework in Fig. 1 consists of two parts: the first part in Fig. 1(a) seeks to learn a representation of the set of observed features $\mathbf{z_1} = f_\theta(\mathcal{O}_t)$ capable of classifying the label $\hat{y} = g_\psi(\mathbf{z_1})$ and estimating the optimal next action $a_t = \pi_\phi(\mathbf{z_1})$. The model has two heads that share the same encoder which leads to improved performance over a model with two separate encoders [Shim et al., 2018]. In parallel, the second network in Fig. 1(b) seeks to learn a related but separate representation $\mathbf{z_2} = f_\chi(\mathcal{O}_t)$, which is fed to a classifier g_ω that predicts the sensitive attribute $\hat{b}$. Crucially and different from prior work on adversarial representation learning, the second adversarial classification task cannot have a shared encoder with the first two tasks as this could encourage the encoder to mask the unfairness of features directly which, in turn, would not lead to selecting a set of fair features that generalize to any downstream task. While in adversarial representation learning the adversarial loss is backpropagated directly through a gradient reversal layer to update the encoder [Goodfellow et al., 2014, Edwards and Storkey, 2016], our agent learns to fool the adversary by selecting the set of features that maximize the adversarial classification loss.

We realize f_θ, g_ψ, f_χ, g_ω and π_ϕ as neural networks parametrized by θ, ψ, χ, ω, and ϕ, which are optimized using alternating gradient descent steps. To facilitate encoding of partially observed feature sets, we adopt a feature-level set encoder [Shim et al., 2018]. Each observed feature x_i is first mapped to a memory vector $\mathbf{m}_i$ after which an LSTM processes the set of memory vectors repeatedly while an attention layer improves the set embedding. The attention step ensures the input is order-invariant. The final set embedding $\mathbf{z}_1$ is fed to both the classifier and the policy network. A second independent set embedding $\mathbf{z}_2$ is fed to the adversary. We refer to App. A for details on the set encoding process and to App. B for implementation details.

Adversarial reward function We compare two different loss functions to compute the rewards for the adversary. First, earlier work on adversarial fair representation learning for demographic parity has shown that using binary cross-entropy (CE) loss for both the classifier and the adversary encourages fair and high-value representations [Edwards and Storkey, 2016, Beutel et al., 2017]

$$\mathcal{L}_A^{CE}(\mathcal{O}_T) = -\left(b\log(g_\omega(f_\chi(\mathcal{O}_T))) + (1-b)\log(1-g_\omega(f_\chi(\mathcal{O}_T)))\right) \tag{5}$$

where the adversary only has access to the final feature set $\mathcal{O}_T$ obtained after stopping. Though effective, $\mathcal{L}_A^{CE}$ fails to account for demographically unbalanced training data. To address this problem, [Madras et al., 2018] introduces group-normalized L_1 (GNL_1) loss as a more natural relaxation of demographic parity, which we adopt to compute the rewards from the adversary

$$\mathcal{L}_A^{GNL_1}(\mathcal{O}_T) = \frac{|P|}{2|P_b|}|g_\omega(f_\chi(\mathcal{O}_T)) - b| \tag{6}$$

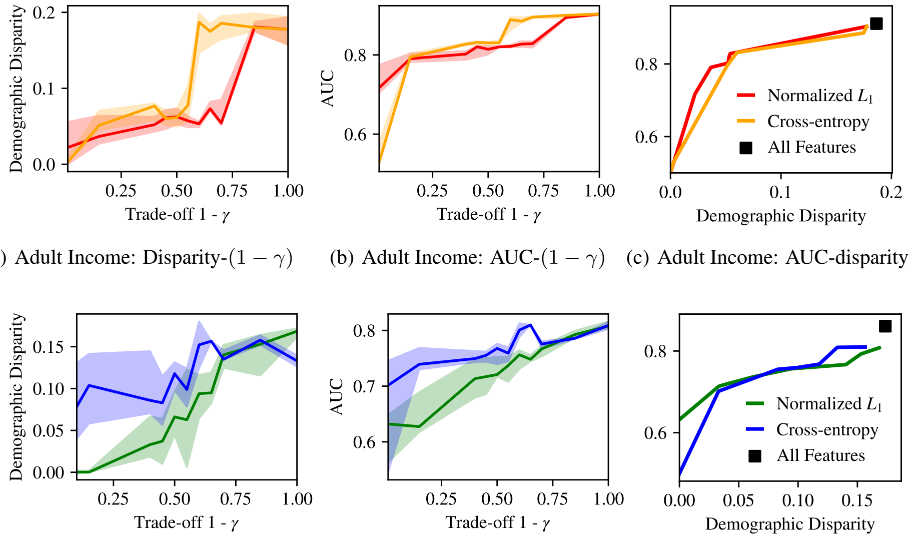
where P_0 and P_1 are the protected subgroups with respectively attributes $b = 0$ and $b = 1$. As neural networks have difficulty learning with L_1 loss [Janocha and Czarnecki, 2017], we continue to use cross-entropy loss to train the adversary but use $\mathcal{L}_A^{GNL_1}$ to compute the final rewards for the agent. We refer to Madras et al. [2018] for the theoretical properties of both loss functions.

4 Experiments

DADI seeks to select the subset of features that can be used by third parties with the assurance that their trained classifiers are both fair and accurate. As exact demographic parity is hard to enforce in practice, we use demographic disparity $|P(\hat{y} = 1 \mid b = 0) - P(\hat{y} = 1 \mid b = 1)|$ as measure for the degree of unfairness. The performance of the classifier is measured using the Area Under the Receiver Operating Characteristics curve (AUC) to account for the imbalanced label distributions.

Datasets We evaluate DADI empirically on the UCI Adult and Mexican Poverty datasets. We use one-hot encoding for categorical features and standardize numerical features. For the mapping to actions, we combine multiple one-hot encoded binary features that stem from the same categorical feature into a single action (e.g. the binary features *marital=divorced, marital=married* and *marital=single* correspond to a single action that acquires these features simultaneously). We use 8-fold cross validation with a random 87.5%/12.5% train-val/test split. We further split the train-val set into training and validation data using a second random 80%/20% split.

The Adult Dataset from UCI Machine Learning Repository [Lichman et al., 2013] comprises 14 demographic and occupational attributes, which translates after preprocessing into 98 continuous



(a) Adult Income: Disparity-$(1-\gamma)$ (b) Adult Income: AUC-$(1-\gamma)$ (c) Adult Income: AUC-disparity

(d) Mex. Poverty: Disparity-$(1-\gamma)$ (e) Mex. Poverty: AUC-$(1-\gamma)$ (f) Mex. Poverty: AUC-disparity

Figure 2: DADI for mitigating demographic disparity across subgroups in the Adult and Mexico datasets. Subfigures (a),(b), (d) and (e) show respectively the AUC and disparity for a range of trade-off parameters $1-\gamma$. The lines are plotted using the median with first/third quantile as confidence area computed using 8-fold CV. Subfigures (c) and (f) show the Pareto front along the AUC-disparity trade-off. The black square represents the baseline unfair classifier for which we use the pretrained classifier together with the full feature set. The median AUC and disparity are again computed across the 8 folds.

and binary features and 14 actions for 48,842 individuals, with the goal of classifying whether a person's income is above \$50,000 (25% are above). Rows with missing values are omitted resulting in a dataset with 45,222 samples. In line with previous work, we use gender as the sensitive attribute, listed as male or female.

The Mexican Poverty dataset is extracted from the Mexican household survey 2016, which contains ground-truth household poverty levels and 99 attributes, related to household information such as the number of rooms or the type of heating system [Ibarrarán et al., 2017]. The processed dataset is obtained from Noriega-Campero et al. [2019] and comprises a sample of 70,305 households in Mexico, with 183 continuous and one-hot encoded binary features and 99 actions. Classification is binary according to the country's official poverty line, with 36% of the households having the label poor. The considered sensitive attribute describes whether the head of the household is a senior citizen or not.

Results Fig. 2 shows the results for both datasets. First, Figs. 2(a),2(b),2(d), and 2(e) show that increasing $1-\gamma$, i.e., decreasing the relative weight of the adversarial reward γ, leads to an increase in both performance and disparity for both choices of adversarial reward functions. Naturally, as the adversarial reward becomes less important, the agent will have a stronger incentive to maximize the accuracy which, in turn, leads to the collection of more features and thus higher AUC at the cost of a higher disparity.

Importantly, however, we observe that while the AUC increases drastically from the start, demographic parity only increases drastically for larger values of $1-\gamma$, allowing for agents that achieve good predictive performance with minimal disparity loss. This conclusion is supported by Figs. 2(c) and 2(f) where we visualize the Pareto front along the AUC-disparity trade-off. These results are encouraging as we show that a data collector can still maintain good performance while only having access to a unique fair subset of features for each data owner. Finally, we observe that the group-normalized L_1 reward generally results in a better trade-off, especially in the most important fairness range for small values of the disparity.

5 Conclusion and Future Work

A number of recent works have focused on adversarially learning fair representations. However, the methods underlying these works, are ineffective when the data owner is required to share raw features, a key aspect in many use cases where features are collected for both human and machine decision making. To tackle this problem, we propose DADI, to our knowledge the first framework for dynamic adversarial discovery of fair information. We frame the data owner's choice as a reinforcement learning problem where an agent selects a subset of features while an adversary critiques potentially unfair feature sets. Experimentally, we demonstrate how our framework guides information discovery for ensuring demographic parity and how it allows the data owner to efficiently trade-off fairness and accuracy.

Importantly, however, our framework is more generally applicable in settings where a data owner may wish to guard itself against a naive or malicious data collector by sharing only a subset of features. First, by changing the adversarial objective function, the framework in [Madras et al., 2018] demonstrates that one can achieve other notions of fairness such as equal opportunity and equal odds. Second, several recent works have formulated adversarial objectives to attain (differentially) private data representations. These objectives could be adopted using DADI to automate dynamic discovery of private information [Yang et al., 2018, Phan et al., 2019] which could be further extended by encoding features in different levels of precision (such as age by year or age by decade), allowing the agent to select the level of precision that maximizes accuracy while minimizing privacy risk. Finally, adding monetary acquisition costs of features as a penalty at each collection step would allow our agent to holistically trade-off accuracy, information costs, and fairness or privacy [Shim et al., 2018].

Acknowledgements

The authors want to thank IBM for access to their compute resources and Prasanna Sattigeri for helpful discussions. AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the CFI.

References

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

Aaron Chalfin, Oren Danieli, Andrew Hillis, Zubin Jelveh, Michael Luca, Jens Ludwig, and Sendhil Mullainathan. Productivity and selection of human capital with machine learning. *American Economic Review*, 106(5), 2016.

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

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *The International Conference on Learning Representations (ICLR)*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, page 3315, 2016.

Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, 33(4):847–856, 2007.

Pablo Ibarrarán, Nadin Medellín, Ferdinando Regalia, Marco Stampini, Sandro Parodi, Luis Tejerina, Pedro Cueva, and Madiery Vásquez. *How Conditional Cash Transfers Work*. Number 8159 in IDB Publications (Books). Inter-American Development Bank, 2017. ISBN ARRAY(0x47b15000). URL `https://ideas.repec.org/b/idb/idbbks/8159.html`.

Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Conference on Theoretical Foundations of Machine Learning (TFML)*, 2017.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *Innovations in Theoretical Computer Science (ITCS)*, 2017.

Balaji Krishnapuram, Shipeng Yu, and R Bharat Rao. *Cost-sensitive Machine Learning*. CRC Press, 2011.

Moshe Lichman et al. Uci machine learning repository, 2013.

John Logan. *Diversity and disparities: America enters a new century*. Russell Sage Foundation, 2014.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243, 2019.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3381–3390, 2018.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Alejandro Noriega-Campero, Michiel Bakker, Bernardo Garcia-Bulle, and Alex Pentland. Active fairness in algorithmic decision making. *Proceedings of AAAI / ACM Conference on Artificial Intelligence, Ethics, and Society*, 2019.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019. ISSN 0036-8075. doi: 10.1126/science.aax2342. URL `https://science.sciencemag.org/content/366/6464/447`.

NhatHai Phan, Ruoming Jin, My T Thai, Han Hu, and Dejing Dou. Preserving differential privacy in adversarial learning with provable robustness. *arXiv preprint arXiv:1903.09822*, 2019.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, pages 1368–1378, 2018.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *The International Conference on Learning Representations (ICLR)*, 2015.

Tsung-Yen Yang, Christopher Brinton, Prateek Mittal, Mung Chiang, and Andrew Lan. Learning informative and private representations via generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1534–1543. IEEE, 2018.

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

A Set Encoder

A set encoder is used to encode arbitrary sets of features. The set encoder was introduced as part of the sequence-to-sequence framework in [Vinyals et al., 2015], while the authors in [Shim et al., 2018] adopt it for active feature-value acquisition. The set encoder has two parts: a *reading block* and a *processing block*. First, each feature is represented by a vector $\mathbf{u}_j = [x_j \; \mathcal{I}(j)]$ where x_j is the feature-value and $\mathcal{I}(j)$ is a one-hot vector with 1 at position j and zeros elsewhere, allowing the network to incorporate coordinate information. The reading block embeds each vector $\mathbf{u}_j$ onto a memory vector $\mathbf{m}_j$ using a neural network with a shared set of parameters across all features $j \in \{1, \ldots, d\}$. The processing block reads the memory (so all memory vectors) into an initial reading vector $\mathbf{r}_0 = \frac{1}{N} \sum_j \mathbf{m}_j$ at processing step 0. This vector $\mathbf{r}_0$ is padded with zeros and fed to an LSTM to compute an initial query vector $\mathbf{q}_0$. At each consecutive time step t an attention weight for each memory vector $\mathbf{m}_i$ is computed using

$$a_{i,t} = \frac{\exp\left(\mathbf{m}_i^T \mathbf{q}_t\right)}{\sum_j \exp\left(\mathbf{m}_j^T \mathbf{q}_t\right)} \tag{7}$$

where $\mathbf{m}_i^T \mathbf{q}_t$ is the dot product of the memory and query vectors. Using the attention vector $\mathbf{a}_t$, we update the reading vector $\mathbf{r}_t = \sum_i a_{i,t} \mathbf{m}_i$ which we concatenate with the query vector and feed to the LSTM to compute the next query vector $\mathbf{q}_{t+1}$ =LSTM($[\mathbf{q}_t \; \mathbf{r}_t]$). In turn, this new query vector is used to compute the new attention vector $\mathbf{a}_t$, We repeat this process for a fixed number of processing steps to achieve a final readout vector $\mathbf{r}_T$, which is subsequently fed to the classifiers and policy network. We refer to [Vinyals et al., 2015] for a more detailed description over the encoder and experiments for different number of processing steps. Note that the attention mechanism guarantees that the final readout vector $\mathbf{r}_T$ retrieved from processing is invariant to different permutations of the features in the set.

B Architecture and training details

Architecture We use two separate encoders f_θ and f_χ with the same architecture but different parameters where one feeds into the label classifier and one feeds into the adversary. The encoders consist of a memory block, a neural network with two hidden layers of 64-64 units that maps each feature value and its coordinate information to a 32-dimensional real-valued memory vector, and a processing block, an LSTM with 32 hidden units that performs 5 processing steps over the memory to obtain a final read vector. Both classifiers g_ψ and g_ω and the policy network π_ϕ are realized as neural networks with two hidden layers of 64-64 units. The networks share the same architecture for both datasets and use rectified linear units (ReLUs) as activation functions.

Pretraining In the first training phase, we train the encoders f_θ and f_χ, and classifiers g_ψ and g_ω with both the full set of features and randomly missing features. To obtain the partially missing feature sets, we drop each feature with probability $p \sim U(0, 1)$, sampled once for instance to encourage different degrees of sparsity across instances. We train the models using the Adam optimizer with binary cross-entropy loss for 10,000 iterations and a batch size of 64. In each batch, half of the samples have randomly missing features and half contain the full feature set. We evaluate the AUC of the models on a validation set with partially missing features and save the models with the highest validation score for joint training.

Joint Training In the second training phase, the policy network π_ϕ and the classifier g_ψ are trained jointly for 10,000 iterations. We use n-step Q-Learning [Mnih et al., 2016] with 4 steps and follow the implementation in [Shim et al., 2018] where multiple agents run in parallel and collect n-step experiences $(s_t, a_t, s_{t+1}, a_{t+1}, ..., s_{t+n}, a_{t+n})$ using ϵ-greedy exploration. We decrease ϵ linearly in the first 5,000 iterations from 1 to 0.1. We train with 64 agents in parallel, one agent for one respective instance in a batch of 64. After collecting a running history of n-step experiences, f_θ,

π_ϕ and g_ψ are jointly updated. The policy network π_ϕ and encoder f_θ are updated using gradient descent by backpropagating the squared loss $(Q(s_t, a_t) - R)^2$ of the estimated Q-values and the target Q-values. Q-values corresponding to actions of already acquired features are manually set to $-\infty$ to prevent the agent from selecting the same feature twice. To account for overestimation of Q-values and improve stability, we use a target Q network $\pi_{\phi'}$, which is a delayed copy of the online policy network π_ϕ, that gets updated every 100 joint training iterations by copying the parameters of π_ϕ. The estimated Q-values are defined as $Q(s_{t+n}, \arg\max_a Q(s_{t+n}, a; \phi); \phi')$. The classifiers g_ψ and g_ω, together with encoders f_θ and f_χ are trained using the collected experiences of the agents. Each state $\{x_j\}_{j \in \mathcal{O}_t}$ in the history of experiences represents a partial feature vector that is used in combination with the true label to update the classifier. In line with the pretraining phase, the classifier is trained using binary cross-entropy loss and the Adam optimizer.

Chapter 5

Discussion

Machine learning systems are used in situations that directly affect human lives, from credit risk scoring, to criminal risk assessment and predictive policing. With the increasing usage of predictive models, there are increasing concerns that their outputs may be supporting decisions that result in systematic discrimination and unfair treatment based on sensitive characteristics such as gender, race or nationality. Hence, it is crucial to address fairness in such systems. This thesis is aimed at contributing to the design of novel methods to mitigate for fairness in a setting where a decision maker can adaptively collect information in a budget-constrained setting before a predictive decision is made. Specifically, two methods were proposed.

Chapter 3 introduces a framework for confidence-based fair stopping during the information collection process. The main idea is to stop acquiring additional features for an individual when a certain confidence in the decision is reached. This stopping criterion is analytically derived based on the sub-group specific base rate and relates to a desirable error rate for that subpopulation. Importantly, this approach is agnostic to the feature acquisition strategy and the classifier, as long as it can handle partial feature vectors. The framework is effectively able to achieve equal opportunity for calibrated probabilistic classifiers without relying on optimization or randomization methods.

In Chapter 4, a framework for the dynamic adversarial discovery of fair information is proposed. In contrast to the previous contribution, this method jointly combines a feature acquisition strategy, a stopping criterion, and classifier in a unified reinforcement learning framework to trade off predictive performance and fairness. This is done by modeling the prediction-time active feature-value acquisition (AFA) setup as a Markov decision process (MDP) and adding an adversarial loss component to the reward function of the agent. In this way, the agent learns to select a subset of features that is predictive for the label but not for the sensitive attribute. This mitigates potential demographic disparities in downstream classification tasks.

A common overarching theme of both contributions is that the information budget is leveraged as an additional degree of freedom to achieve fairness objectives. In contrast to the work in [11], the final feature sets are personalized and not static across subgroups, meaning that each individual can have different selected features and a different amount of features. This allows to cater for individual-level nuances and enables a more efficient feature acquisition process. Both frameworks were effectively evaluated using real-world data sets and demonstrated abilities to achieve different popular notions of group fairness, i.e., demographic parity, equal opportunity, and calibration by group.

Holistically, this work presents some interesting perspectives at the intersection of fairness in machine learning and AFA systems, and is relevant in many real-world settings such as social targeting, criminal justice, healthcare resource allocation or credit risk scoring. Especially in situations where decision makers need to make decisions under resource constraints, e.g. time, cost or information, the presented methods provide a toolset to achieve popular notions of statistical fairness. For example, in public pandemic crises decisions need to be made how to allocate intensive care unit (ICU) spots based on e.g. the estimated risk of fatality of individual cases. The estimated risk of fatality is a function of the information (features) that were acquired per case (e.g. reported symptoms, lab test results). However, the global information budget is constrained by the capacity of the healthcare system (e.g. number of healthcare workers or possible lab tests per day) and a decision maker needs to decide who will be tested or gets further treatment. In this context, a decision maker needs to gather the right information on a case-level to make confident decisions (who gets an ICU bed), while also considering fair access to the healthcare system across social classes in the population.

5.1 Limitations and Future Work

One limitation we experienced during the experiments is that the data sets need to carry a large number of features that are predictive for the task. For the method introduced in chapter 3, it could otherwise happen that the confidence thresholds are not reached even though all available features have been queried. Also for the RL framework in chapter 4, feature-rich data sets worked better as they provide the agent with more flexibility to choose between features that are differently informative for the label and sensitive attribute. Hence, the agent can achieve a more granular trade-off between fairness and predictive performance. As most data sets are not collected for the purpose of AFA, this limitation held in the experimental evaluation of the approaches. However, this does not apply in real-world AFA situations, where further features can be acquired at a cost at prediction-time regardless of how many features were observed in the training set.

While the focus of both proposed methods lies on fairness objectives, we encourage future research to explore extensions to incorporate feature-specific costs and privacy objectives. First, we assumed a simplified setting where features come at uniform cost. However, in practice, the acquisition cost of different features can vary. Second, intuitively, we concluded that using only a subset instead of all available features per individual should provide more privacy. However, explicitly incorporating privacy as a cost, mitigating privacy disparities across groups or connecting it to the line of work of differential privacy would be possible paths moving forward. Despite some interesting work at the intersection of group fairness and individual fairness [76, 77], most methods are still difficult to apply in practice. The work in Chapter 3 can be considered as an attempt to stake a ground between group and individual fairness, as it provides confident decisions on an individual level while also equalizing group-level statistics. We want to encourage future research to investigate the individual-level fairness guarantees and explore new methods to achieve individual fairness.

5.2 Conclusion

In conclusion, all efforts at the frontier of fairness in machine learning are already an important step forward. However, in practice, the adoption of these methods still poses challenges [78]. It is important to highlight that in order to bring fair algorithms from research into practice and make a real-world impact, a dialogue between many stakeholders such as politicians, ethicists, legal experts, researchers, and industry practitioners is necessary. With the right regulatory framework, a selection of fairness tools and a *fairness-by-design* product mindset, we can take the right steps to move towards a more inclusive society while still taking advantage of machine learning systems.

Bibliography

[1] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *ArXiv*, abs/1501.02876, 2015.

[2] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[3] S.M. McKinney, M. Sieniek, and V. Godbole. International evaluation of an ai system for breast cancer screening. *Nature*, 577:89–94, 1968.

[4] Andrew Ng. How to choose your first ai project. *HBR*, 2019.

[5] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: There is software used across the country to predict future criminals. and it is biased against blacks. *ProPublica*, 2016.

[6] Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*, 2018.

[7] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, New York, NY, USA, 23–24 Feb 2018. PMLR.

[8] Russell Brandom. Facebook has been charged with housing discrimination by the us government. *The Verge*, 2019.

[9] Natasha Lomas. Blackbox welfare fraud detection system breaches human rights, dutch court rules. *Techcrunch*, 2020.

[10] Maytal Saar-Tsechansky, Prem Melville, and Foster Provost. Active feature-value acquisition. *Management Science*, 2006.

[11] Alejandro Noriega-Campero, Michiel Bakker, Bernardo Garcia-Bulle, and Alex Pentland. Active fairness in algorithmic decision making. *Proceedings of AAAI / ACM Conference on Artificial Intelligence, Ethics, and Society*, 2019.

[12] Giuseppe Casalicchio. *On Benchmark Experiments and Visualization Methods for the Evaluation and Interpretation of Machine Learning Models.* PhD thesis, Ludwig-Maximilians-Universitaet Muenchen, 2019.

[13] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[14] Eduardo P. Costa, Ana C. Lorena, Andre C. P. L. F. Carvalho, and Alex A. Freitas. A review of performance evaluation measures for hierarchical classifiers. In *Proceedings of the 2007 AAAI Workshop Evaluation Methods for Machine Learning II. AAAI*, 2007.

[15] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[16] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass prob- ability estimates. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002.

[17] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthey Weather Review*, 78(1):1–3, 1950.

[18] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[19] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[20] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* 2009.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[22] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICMLâ10, page 807â814, Madison, WI, USA, 2010. Omnipress.

[23] Facundo Bre, Juan Gimenez, and VÃctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 11 2017.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating error. *Nature*, 323:533, 1986.

[25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[26] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees.* 1984.

[27] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[28] Olivier Pauly. *Random Forests for Medical Applications.* PhD thesis, Technische UniversitÃt MÃ$\frac{1}{4}$nchen, 2012.

[29] Alexander Amini. Deep reinforcement learning mit 6.s191 lecture 5, 2019.

[30] Marco Wiering and Martijn van Otterlo. *Reinforcement Learning: State-of-the-Art*, volume 12. Springer, 2012.

[31] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, 2nd edition, 2018.

[32] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[33] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[35] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2015.

[36] Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

[37] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2015.

[38] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *ICML*, 2015.

[39] George Gorry and Octo Barnett. Sequential diagnosis by computer. 1968.

[40] Tianshi Gao and Daphne Koller. Active classification based on value of classifier. In *Advances in Neural Information Processing Systems*, 2011.

[41] Pallika Kanani and Prem Melville. Prediction-time active feature-value acquisition for cost-effective customer targeting. *Advances In Neural Information Processing Systems (NIPS)*, 2008.

[42] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of machine learning research*, 8(Jul):1623–1657, 2007.

[43] Itamar Reis, Dalya Baron, and Sahar Shahaf. Probabilistic random forest: A machine learning algorithm for noisy data sets. *The Astronomical Journal*, 157(1):16, 2018.

[44] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[45] Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, pages 1368–1378, 2018.

[46] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018.

[47] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[48] Balaji Krishnapuram, Shipeng Yu, and R Bharat Rao. *Cost-sensitive Machine Learning.* CRC Press, 2011.

[49] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[50] Mohammad Kachuee, Orpaz Goldstein, Kimmo Kärkkäinen, Sajad Darabi, and Majid Sarrafzadeh. Opportunistic learning: Budgeted cost-sensitive learning from data streams. *CoRR*, abs/1901.00243, 2019.

[51] Jaromír Janisch, Tomás Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. *CoRR*, abs/1711.07364, 2017.

[52] Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, 33(4):847–856, 2007.

[53] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. Prediction policy problems. *American Economic Review*, 105(5), 2015.

[54] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint*, 2016.

[55] Akhil Kodiyan. An overview of ethical issues in using ai systems in hiring with a case study of amazon's ai based hiring tool. 11 2019.

[56] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

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

[58] Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffry Svacha, and Madeleine Udell. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *FAT* '19*, 2019.

[59] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

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

[61] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *CoRR*, abs/1609.05807, 2016.

[62] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5 2:153–163, 2016.

[63] Solon Barocas and Moritz Hardt. Tutorial on fairness in machine learning. Conference on Neural Information Processing Systems, 2017.

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

[65] Flávio du Pin Calmon, Dennis Wei, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized data pre-processing for discrimination prevention. *ArXiv*, abs/1704.03354, 2017.

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

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

[68] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018.

[69] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment disparate impact: Learning classification without disparate mistreatment. In *WWW*, 2017.

[70] Xinyue Shen, Steven Diamond, Yuantao Gu, and Stephen P. Boyd. Disciplined convex-concave programming. *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1009–1014, 2016.

[71] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint*, 2017.

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

[73] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *FAT* '19*, 2018.

[74] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

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

[76] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv*, 2017.

[77] Úrsula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1944–1953, 2018.

[78] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miroslav Dudík, and Hanna M. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? *ArXiv*, abs/1812.05239, 2019.