Promoting Fairness in Learned Models by Learning to Active Learn under Parity Constraints

Abstract

Machine learning models can have consequential effects, and disparities in error rate can lead to harms suffered more by some groups than others. Past algorithmic approaches mitigate such disparities for fixed training data; we ask: what if we can gather more data? We develop a meta-learning algorithm for parity-constrained active learning that learns a policy to decide which labels to query so as to maximize accuracy subject to parity constraints, using forward-backward splitting at the meta-learning level. Empirically, across three classification tasks and different parity metrics, our approach outperforms alternatives by a large margin.

1 Introduction

Machine learning models often lead to harms due to disparities in behavior across social groups: an automated hiring system may be more likely to recommend hiring people of privileged races, genders, or age groups (Wachter-Boettcher 2017; Giang 2018). These disparities are typically caused by biases in historic data (society is biased); a substantial literature exists around "de-biasing" methods for algorithms, predictions, or models (, i.a.). Such approaches always assume that the training data is fixed, leading to a false choice between efficacy (e.g., accuracy, AUC) and "fairness" (typically measured by a metric of parity across subgroups (Chen, Johansson, and Sontag 2018; Kallus and Zhou 2018)). This is in stark contrast to how machine learning *practitioners* address disparities in model performance: they *collect more data* that's more relevant or representative of the populations of interest (Veale and Binns 2017; Holstein et al. 2019). This disconnect leads to a mismatch between sources of bias, and the algorithmic interventions developed to mitigate them (Zarsky 2016).

i.a. means "amongst others"

We consider a different trade-off: given a pre-existing dataset, which may have been collected in a highly biased manner, how can we manage an efficacy vs *annotation cost* trade-off under a target parity constraint? We call this problem *parity-constrained active learning*, where a maximal disparity (according to any of a number of different measures, see Table 1) is enforced during a data collection process. We specifically consider the case where some "starting" dataset has already been collected, distinguishing our procedure from more standard active learning settings in which we typically start from no data ((Settles 2009), see §2). The goal then is to collect as little data as is needed to keep accuracy high while maintaining a constraint on parity (as a measure of fairness). As an example, consider disparities in pedestrian detection by skin tone (Wilson, Hoffman, and Morgenstern 2019): A pedestrian detector is trained based on a dataset of 100k images, but an analysis shows that it performs significantly better at detecting people with light skin than people with dark skin. Our goal is to label few *additional* samples while achieving a high accuracy under a constraint on the disparity between these groups.[1]

experiments around this would be interesting

not that convincing

We propose to solve the parity-constrained active learning problem using a meta-learning approach, very much in the style of recent work on meta-learning for active learning (Konyushkova, Sznitman, and Fua 2017; Bachman, Sordoni, and Trischler 2017; Fang, Li, and Cohn 2017). Our algorithm, PARITY-CONSTRAINED META ACTIVE LEARNING (PANDA; see §3), uses data to learn a selection policy that picks which examples to have labeled. The data on which it learns this selection policy is the pre-existing (possibly biased!) dataset from which it will continue learning.

Tenuous at best

To achieve this, PANDA simulates many parity-constrained active learning tasks on this pre-existing dataset, to learn the selection policy. Technically, PANDA formulates the parity-constrained active learning task as a bi-level optimization problem. The inner level corresponds to training a classifier on a subset of labeled examples. The outer level corresponds to updating the selection policy choosing this subset to achieve a desired fairness and accuracy behavior on the trained classifier. To solve this constrained bi-level optimization problem, PANDA employs the *Forward-Backward Splitting* (FBS, (Lions and Mercier 1979; Combettes and Wajs 2005; Goldstein, Studer, and Baraniuk 2014)) optimization method (also known as the proximal gradient method). Despite its apparent simplicity, FBS can handle non-differentiable objectives with possibly non-convex constraints while maintaining the simplicity of gradient-descent methods.

Through exhaustive empirical experiments (§4), we show the following:

doubt

[1] Code: https://bit.ly/34eVrsG

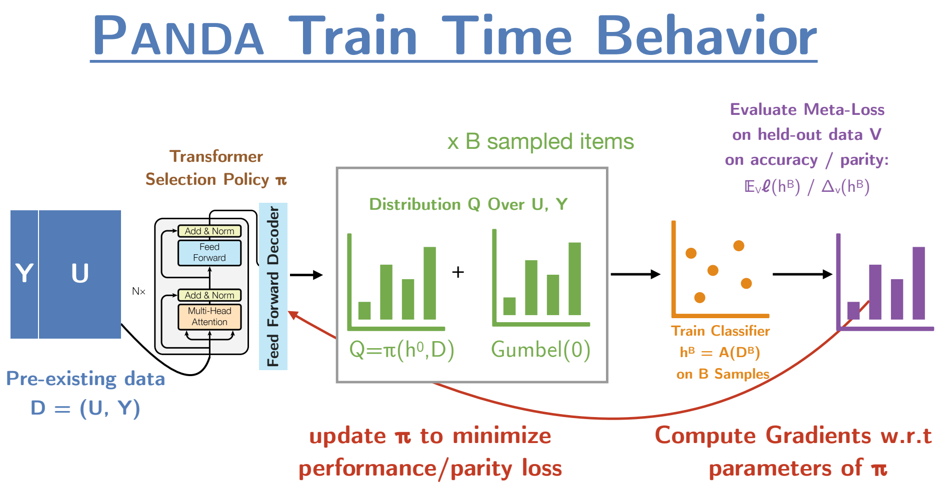

Figure 1: Train time behavior of PANDA. The figure shows a training step of PANDA. The model of interaction is similar to **??**, however, at training time, we also have access to the labels Y for simulating the parity-constrained active learning setting. We model the selection policy π using a transformer encoder followed by a feed-forward decoder. Each layer in the transformer encoder has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The model is trained end-to-end where a Gumbel-Softmax reparameterization trick is used to avoid back-propagating through the sampling procedure from the distribution Q.

1. PANDA is effective: it outperforms alternative active learning algorithms by a large margin under the same setting while enforcing the desired behavior on fairness.
2. PANDA is general-purpose: it learns the selection policy end-to-end and can handle a wide set of non-differentiable and non-convex constraints on fairness parity using Gumbel-Softmax reparameterization (Gumbel 1948; Jang, Gu, and Poole 2016; Maddison, Mnih, and Teh 2016) and differentiable approximations.
3. PANDA is powerful: it employs a Transformer model architecture (Vaswani et al. 2017) to represent the learned selection policy. This architecture has achieved state-of-the-art performance in many tasks including language modeling (Dai et al. 2019), machine translation (Dehghani et al. 2018), and unsupervised pre-training (Devlin et al. 2018).

2 Background and Related Work

Concerns about biased or disparate treatment of groups or individuals by computer systems has been raised since the 1990s (Friedman and Nissenbaum 1996). Machine learning and other statistical techniques provide ample opportunity for pre-existing societal bias to be incorporated into computer body systems through data, leading to a burgeoning of research studying disparities in machine learning (Abdollahi and Nasraoui 2018; Crawford and Calo 2016, i.a.). Arguably, because society is biased, societal data will be biased, and therefore, if unchecked, any machine learning model trained on such data will inherit its biases.

Much technical machine learning research has gone into defining what disparate treatment means formally, leading to a zoo of parity metrics (Narayanan 2018) (see Table 1 for examples), proofs of their incompatibilities (Chouldechova 2017; Kleinberg, Mullainathan, and Raghavan 2016), and analyses of how they conform to normative notions of fairness (Srivastava, Heidari, and Krause 2019). This has led to machine learning algorithms that optimize not just for accuracy, but rather for accuracy subject to a constraint on parity between known groups (Agarwal et al. 2018).

A parallel line of research has considered the human side of analyzing disparities in machine learning systems, including visualization (Cabrera et al. 2019), debugging (Chen, Johansson, and Sontag 2018), and needs-finding (Veale and Binns 2017; Holstein et al. 2019). One finding of the latter is that machine learning practitioners and data scientists often have control over training data, which is not taken into account in most technical machine learning research. For instance, (Holstein et al. 2019)'s results show that 78% of practitioners who had attempted to address disparities did so by trying to collect more data, despite the lack of tools that support this.

Curating more data is not a foreign concept in the machine learning research: active learning—the learning paradigm in which the learner itself chooses which examples to have labeled next—has been studied extensively over the past five decades (Settles 2009; Fedorov 2013; Angluin 1988; Cohn, Atlas, and Ladner 1994; Jiang and Ip 2008). Most active learning approaches select samples to label based on some notion of uncertainty (e.g., entropy of predictions). Most relevant to our work are recent active learning approaches based on meta-learning (Bachman, Sordoni, and Trischler 2017; Fang, Li, and Cohn 2017): here, instead of designing the selection strategy by hand, the selection strategy is learned based on offline, simulated active learning problems. So long as those offline problems are sufficiently similar to the target,

Metric	Description & Mathematical Definition
Demographic Parity	Predictions $h(x)$ are statistically independent of the group $g(x)$ (Feldman et al. 2015): $\Delta^{\mathrm{DP}}(h) \triangleq \max_a \ \lvert \mathbb{E}[h(x) \mid g(x) = a] - \mathbb{E}[h(x)] \rvert$
Equalized Odds	Predictions $h(x)$ are independent of the group $g(x)$ given the true label y (Hardt, Price, and Srebro 2016): $\Delta^{\mathrm{EO}}(h) \triangleq \max_{a,y} \ \lvert \mathbb{E}[h(x) \mid g(x) = a, Y = y] - \mathbb{E}[h(x) \mid Y = y] \rvert$
Error-rate Balance	False positive and false negative error rates are equal across groups (Chouldechova 2017): $\Delta^{\mathrm{EB}}(h) \triangleq \max_{a,a',y} \ \lvert \mathbb{E}[h(x) \mid g(x) = a, Y = y] - \mathbb{E}[h(x) \mid g(x) = a', Y = y] \rvert$

Table 1: Three common measures of disparity for binary classification (extensions to multiclass are generally straightforward), expressed in terms of differences in expected values of predictions (where we take $h : \mathcal{X} \to \{0, 1\}$). We denote by $g(x)$ the group to which the example x belongs. In some work, disparities are taken to be *ratios* of expectations, rather than differences.

real, active learning problem, there is hope that the learned strategy will generalize well.

We are aware of only one paper that considers active learning in the context of fairness: Fair Active Learning (FAL) by (Anahideh and Asudeh 2020). FAL uses a handselection strategy that interpolates between an uncertainty-based selection criteria, and a "fairness" criteria that estimates the impact on disparity if the label of a given point were queried (by computing expected disparity over all possible labels). FAL selects data points to be labeled to balance a convex combination of model accuracy and parity, with the trade-off specified by a hyperparameter. Empirically, (Anahideh and Asudeh 2020) showed a significant reduction in disparity while maintaining accuracy. Our setting is slightly different than FAL—we assume pre-existing data—but we compare extensively to this approach experimentally under similar conditions (§4).

3 Problem Definition and Proposed Approach

In this section we define *parity-constrained active learning* and describe our algorithm, PANDA.

Problem Definition: Parity-Constrained Active Learning

$\mathcal{H} \subset \mathcal{Y}$ H is a subset of Y

$A : (x, y) \to h$, $h(x) \to \hat{y}$

We consider the following model. We have collected a dataset of N *labeled* examples, $D^0 = (\boldsymbol{x}_n, y_n)_{n=1}^N$ over an input space $\mathcal{X}$ (e.g., images) and output space $\mathcal{Y}$ (e.g., pedestrian bounding boxes), and have access to a collection of M-many *unlabeled* examples, $U = (\boldsymbol{x}_m)_{m=1}^M$. Each input example $x \in \mathcal{X}$ is associated with a unique group $g(x)$ (e.g., skin tone). We fix a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ and learning algorithm $\mathcal{A}$ that maps a labeled sample D to a classifier $h \in \mathcal{H}$. Finally, we have a loss function $\ell(y, \hat{y}) \in \mathbb{R}^{\geq 0}$ (e.g., squared error, classification error, etc.) and a target *disparity metric*, $\Delta(h) \in \mathbb{R}^{\geq 0}$ (such as those in Table 1). The goal is to label as few images from U as possible to learn a classifier h with high accuracy subject to a constraint that $\Delta(h) < \rho$ for a given threshold $\rho > 0$. We assume access to a (small) validation set V of labeled examples (which can be taken to be a subset of D). We will denote population expectations and disparities by $\mathbb{E}$ and Δ, respectively, and their estimates on a finite sample by $\hat{\mathbb{E}}_A$ and $\hat{\Delta}_A$, where A is the sample.

The specific interaction model we assume is akin to standard active learning with labeling budget B:

1: train the initial classifier on the pre-existing dataset: $h^0 = \mathcal{A}(D^0)$.
2: **for** round $b = 1 \ldots B$ **do**
3: generate categorical probability distribution $Q = \pi(h^{b-1}, U)$ over U using policy π.
4: sample an unlabeled example $\boldsymbol{x} \sim Q$, query its label y, and set $D^b = D^{b-1} \cup \{(\boldsymbol{x}, y)\}$.
5: train/update classifier: $h^b = \mathcal{A}(D^b)$.
6: **end for**
7: **return** classifier h^B, it's validation loss $\hat{\mathbb{E}}_V \ell(y, h^B(\boldsymbol{x}))$ and validation disparity $\hat{\Delta}_V(h^B)$.

Under this model, the active learning strategy is summarized in the example selection policy π. For example, margin-based active learning (Roth and Small 2006) can be realized by setting $\pi(h, U)$ to assign a $Q(\boldsymbol{x}) = \mathbf{1}[\boldsymbol{x} = \boldsymbol{x}^\star]$ where $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x} \in U} \lvert h(\boldsymbol{x}) \rvert$, where h returns the margin. The goal in parity constraint actively learning is to construct a π with minimal expected loss subject to the constraint that $\Delta(h) < \rho$. Just draw the sample with the smallest margin.

PANDA: Learning to Actively Learn under Parity Constraints

We develop a meta-learning approach, PANDA, to address the parity-constrained active learning problem: the selection policy π is trained to choose samples that, if labeled, are likely to optimize accuracy subject to a parity constraint. This learning happens on D itself: by simulating many possible ways additional data could be selected on the historic data, PANDA learns how to select additional examples, even if D itself was sampled in a biased manner.

To learn π, we construct a distribution of meta-training tasks, $\mathfrak{M}$; samples $(L, V) \sim \mathfrak{M}$ consist of a labeled dataset L (to simulate unlabeled data U) and a validation set V. We form $\mathfrak{M}$ by repeatedly reshuffling D, and produce a finite sample of meta-training tasks $\mathfrak{D}$ i.i.d. from $\mathfrak{M}$. The meta-learning problem is then to optimize π on $\mathfrak{D}$ to achieve high accuracy subject to a constraint on disparity. We begin by first writing the parity-constrained problem according to its characteristic function:

$$\hat{h} \in \underset{h \in \mathcal{H} \,:\, \hat{\Delta}_V(h) < \rho}{\operatorname{argmin}} \hat{\mathbb{E}}_V \, \ell(\boldsymbol{y}, h(\boldsymbol{x})) \iff$$

$$\hat{h} \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} \, \hat{\mathbb{E}}_V \, \ell(\boldsymbol{y}, h(\boldsymbol{x})) + \chi_{\hat{\Delta},\rho,V}(h) \quad (1)$$

where $\chi_{\Delta,\rho,V}(h) = 0$ if $\hat{\Delta}_V(h) < \rho$ and $+\infty$ otherwise; for brevity we write $\chi(h)$ when $\hat{\Delta}, \rho, V$ is clear from context. Under reasonable assumptions, both minimizers are finite.

Given this, the meta-learning optimization problem is:

$$\min_{\pi \in \Pi} \hat{\mathbb{E}}_{(L,V)\sim\mathfrak{D}} \left[\hat{\mathbb{E}}_V \ell(\boldsymbol{y}, h_L^\pi(\boldsymbol{x})) + \chi(h_L^\pi) \right] \quad (2)$$

$$\text{where } h_L^\pi = \textsc{ActiveLearnSim}(\mathcal{A}, D, L, \pi) \quad (3)$$

Here, $\textsc{ActiveLearnSim}(\mathcal{A}, D, L, \pi)$ is the interactive algorithm in §3, where U is taken to be L (with labels hidden) and when a label is queried, it is retrieved from L.

When $\mathcal{A}$ is, itself, an optimizer—as it is in most machine learning settings—then formulation Eq 3 is a constrained bilevel optimization problem. The outer optimization is over the sampling policy π, and the inner optimization is over the optimization over h in $\textsc{ActiveLearnSim}$. We assume that $\mathcal{A}$ can be written as a computational graph, in which case the outer objective can be optimized by unrolling the computational graph of $\mathcal{A}$. This introduces second-order gradient terms, but remains computationally feasible so long as the unrolled graph of $\mathcal{A}$ is not too long: we ensure this by only running a few steps of SGD inside $\mathcal{A}$ and using a simple hypothesis class for $\mathcal{H}$.

A is an optimizer, how can it be represented as computational graph?

There remain two challenges to solve Eq 3. The first is to address the discontinuous nature of the characteristic function χ; we use forward-backward splitting to address this. The second is that the unrolling of $\textsc{ActiveLearnSim}$ yields a computational graph that has stochasticity (due to the sampling of unlabeled examples); we use the Gumbel reparameterization trick to address this.

Forward-Backward Splitting (FBS) is a class of optimization methods (Lions and Mercier 1979), which provide the simplicity of gradient descent methods while being able to enforce possibly non-differentiable constraints. In FBS, the objective is separated into a differentiable part $f(x)$ and an arbitrary (not even necessarily smooth) part $g(x)$. The algorithm operates iteratively by first taking a gradient step just with respect to f to an intermediate value: $x' = x - \eta \nabla f(x)$. Next, it computes a proximal step that chooses the next iterate x to minimize $\eta g(x) + ||x - x'||^2/2$. When f is convex, FBS converges rapidly; for non-convex problems (like Eq 3), theoretical convergence rates are unknown, but the algorithm works well in practice.

To apply FBS to our problem, we choose our policy class Π to be a differentiable neural network (see §3). We set f to be the expected loss term in Eq 3, and g to be the characteristic function on the disparity. The gradient step with respect to f can be computed by automatic differentiation of the unrolled computational graph. The proximal step requires projecting onto χ; for complex Π there is no closed-form solution; instead, we run a separate approximate projection step by running several stops of gradient descent on a smoothed version of χ, which takes values 0 when the constraint is satisfied, and takes value $\hat{\Delta}_V(h)$ otherwise. This remains non-continuous, but (sub)differentiable—empirically, this approximate projection always finds an iterate that satisfies the original constraint.

Gumbel Reparameterization is a generic technique to avoid back-propagating through stochastic sampling nodes in the computational graph (Gumbel 1948; Jang, Gu, and Poole 2016; Kool, Van Hoof, and Welling 2019; Maddison, Mnih, and Teh 2016). This trick allows us to sample from the categorical distribution Q by independently perturbing the log-probabilities Q_i with Gumbel noise and finding the largest element, thus enabling end-to-end differentiation through $\textsc{ActiveLearnSim}$, so long as $\mathcal{A}$ is differentiable.

The **Full Training Algorithm** for $\textsc{Panda}$ is summarized in Algorithm 1. Following the Forward-Backward Splitting template, $\textsc{Panda}$ operates in an iterative fashion. Over iterations, $\textsc{Panda}$ simulates a parity-constrained active learning setting for the current model parameters θ^k. Line 4 performs a simple forward gradient descent step to maximize the classifier performance. This step begins at iterate θ^k, and then moves in the direction of the (negative) gradient of the performance loss, which is the direction of steepest descent. Line 5 is the proximal (or backward) step, which enforces the parity constraint; this works even when the parity metric is non-differentiable. In both the gradient descent step and the proximal step, $\textsc{Panda}$ performs bilevel optimization. For example, the gradient step is a gradient with respect to the parameters of the selection policy, of the computational graph defined by $\textsc{ActiveLearnSim}$. That function, itself, performs an optimization of the classifier h.

Network Structure of Selection Policy

The selection policy π takes as input the current classifier h and unlabeled dataset U, and produces a distribution Q over elements of U. We explore policies that are *agnostic* to changes in h, meaning that Q at round b is identical for all b. This introduces a limitation that π cannot directly adapt to changes in h; however, since π is optimized end-to-end, we empirically found this to be a minor limitation. A significant advantage of this choice is that it means that $\textsc{ActiveLearnSim}$ can be unrolled much more easily: the simple Gumbel softmax can be replaced with Gumbel-top-B (Vieira 2014; Kool, Van Hoof, and Welling 2019) and unrolled in a single step, rather than a sequence of B-many steps.

Because π must effectively make all selections in a single step, it is important that π consider each $\boldsymbol{x}$ in the context of all other items in U, and not consider each $\boldsymbol{x}$ individually. We implement this using a Transformer architecture (Vaswani et al. 2017), in which a self-attention mechanism essentially combines information across different $\boldsymbol{x}$s in U. Specifically, we treat the examples in U as an unordered sequence as input to the Transformer encoder[2]. The Transformer architecture

[2]Recall that although Transformers are typically used for *ordered* problems like language modeling (Dai et al. 2019) and machine translation (Dehghani et al. 2018), this is not how they "naturally" work: ordered inputs to Transformers require additional "position" tags.

Algorithm 1 Parity-constrained Active Learning via PANDA

Input: pre-existing datasets D, budget B, loss function ℓ, disparity metric Δ, threshold ρ, meta-learning learning rate schedule $\langle \eta^k \rangle_{k \geq 1}$, and inner learning rate η'

Output: Selection policy π

Initialize selection policy $\pi(\cdot; \theta^0)$ parameterized by θ^0

for iteration $k = 1 \ldots$ convergence **do**

  Split D into pool L and validation set V

  $\hat{\theta}^{\cancel{k+1}} = \theta^k - \eta^k \nabla_\theta \hat{\mathbb{E}}_V \ell\big(y, \textsc{ActiveLearnSim}(\mathcal{A}, D, L, \pi(\cdot; \theta^k)(\boldsymbol{x})\big)$

  $\theta^{\cancel{k+1}} = \mathrm{argmin}_\theta\, \eta^k \chi_{\hat{\Delta},\rho,V}\big(\textsc{ActiveLearnSim}(\mathcal{A}, D, L, \pi(\cdot; \theta))\big) + 1/2 \|\theta - \hat{\theta}^{\cancel{k+1}}\|^2$

end for

return $\pi(\cdot; \theta^{\text{final}})$

function $\textsc{ActiveLearnSim}(\mathcal{A}, D, L, \pi)$

  Let $\langle \boldsymbol{x}_i, y_i \rangle_{i=1}^{|L|}$ be an indexing of L

  for $b = 1 \ldots B$ **do**

    set $\tilde{Q}_i = \pi(h^{b-1}, \boldsymbol{x}_i) + \textsc{Gumbel}(0)$ for all i and pick $j = \arg\max_i \tilde{Q}_i$

    take (a/several) gradient step(s) of the form: $h^b = h^{b-1} - \eta \nabla_h \ell(y_j, h(\boldsymbol{x}_j))$

  end for

  return h^B

end function

employs several layers of self-attention across U with independent feed-forward networks for each position. The final layer of the Transformer can be interpreted as a contextual representation for each $\boldsymbol{x} \in U$, where the context is "the rest of U." We use a final linear softmax layer to map these contextual representations to the probability distribution Q.

Because this model architecture is flexible, it is also data-hungry, and training all of its parameters based just on a small set of B examples is unlikely to be sufficient. This is where the initial dataset D^0 comes in: we pretrain the parameters of the Transformer on D^0 and use the B-many actively selected samples to fine-tune the final layer, thus keeping the required sample complexity small.

4 Experiments

We conduct experiments in the standard active learning manner: pretend that a labeled dataset is actually unlabeled, and use its labels to answer queries. Experimentally, given a complete dataset, we first split it 50/50 into meta-training and meta-testing sections. We use meta-training to pretrain the Transformer model (see §3), and also to train PANDA. All algorithms us the same Transformer representation. The meta-testing section is split again 50/50 into the "unlabeled" set and the test set.

The transformer is not the model

Picking a good dataset for parity-constrained active learning is challenging: it needs to contain a protected attribute, be sufficiently large that an active sample from unlabeled portion is representative (i.e., as the size of the sample approaches the size of the unlabeled data, all algorithms will appear to perform identically), and be sufficiently rich that learning does not happen "too quickly."

We considered three standard datasets: COMPAS (Angwin et al. 2016), Adult Income (Dua and Graff 2017), and Law School (Wightman 1998). Law School has only two features and we found only a few examples are needed to learn; and COMPAS we found to be too small. COMPAS consists of just under $8k$ samples, so after two splits, each set contains only $2k$ samples. We anticipated that this would be too small for three reasons. First, with a budget $B = 400$, many algorithms will end up sampling very similar sets, resulting in difficulties telling approaches apart. Second, we found that after pre-training, 15–20 completely random samples suffice to learn a classifier that is as good as one trained on all the remaining data. Nonetheless, we performed experiments on COMPAS for all baselines and found that while PANDA can fit the meta-training data well, and this generalizes well with respect to *loss*, it has poor generalization with respect to *disparity*. We also ran Fairlearn (described below) on this dataset randomly sampled subsets of the training data, and found that, while it eventually is able to achieve a target disparity level of 0.04 once $B = 400$, at any point with $B < 300$ the test-time disparity is significantly larger. We therefore drop COMPAS from consideration; it seems ill-suited to a warm-start active learning paradigm. This left only the Adult Income dataset for experiments. This dataset consists of $48,842$ examples and 251 features (with one-hot encodings of categorical variables) and the binary prediction task is whether someone makes more than 50k per year, with binary gender as the group attribute (the dataset does not contain information about gender beyond male/female).

I'd expect the COMPAS discussion in the Supp Mat.

Baseline Active Learning Approaches

Our experiments aim to determine how PANDA compares to alternative active learning strategies, including those that explicitly take disparity into account as well as those that do not. Among those that do not consider disparity, we compare to:

Random Sampling – select examples to label uniformly at random.

Margin Sampling – uncertainty-based active learning that

Not reported as it didn't do well

selects the example closest to the current decision boundary (Roth and Small 2006).

Entropy Sampling – uncertainty-based active learning that selects the example with highest entropy of predicted label (Shannon 1948; Settles 2009).

We also consider alternate approaches that take groups and/or disparity into account explicitly.

Group Aware Random Sampling – select examples to label uniformly at random, restricted to the group on which worse performance is achieved by h^0. Closely related to active learning in domain adaptation (Shi, Fan, and Ren 2008; Rai et al. 2010; Wang, Huang, and Schneider 2014).

Fair Active Learning – the fair active learning approach described in §2 that optimizes an interpolation between Entropy Sampling and expected disparity.

Fairlearn – select examples to label uniformly at random, and the run fairlearn to train a classifier to optimize accuracy subject to a parity constraint (Agarwal et al. 2018).

Implementation Details and Hyperparameter Tuning

We use the Transformer Model (Vaswani et al. 2017) implemented in PyTorch (Paszke et al. 2019). We use the standard transformer encoder with successive encoder layers. Each layer contains a self-attention layer, followed by a fully connected feed-forward layer. We use the feed-forward layer for decoding, where we sample B items from the predicted probability distribution in a single decoding step. To ensure a fair-comparison among all approaches, we use the same Transformer architecture as a feature extractor for all approaches. This ensures that PANDA has no additional advantage by observing more training data.

The model is optimized with Adam (Kingma and Ba 2014). We optimize all hyper-parameters with the Bayes search algorithm implemented in comet.ml using an adaptive Parzen-Rosenblatt estimator. We search for the best parameters for learning rate (10^{-2} to 10^{-7}), number of layers in the transformer encoder $(1, 3, 5)$, embedding dimensions for the encoder hidden-layer $(16, 32, 64)$, as well as the initial value for the Gumbel-Softmax temperature parameter (1 to 10^{-6}) which is then learned adaptively as meta-training progresses. The sampled examples are used to train a linear classifier, again we optimize the hyper-parameters for the learning rate and batch size using Bayes search. For active learning model selection, we sweep over parameters using the random sampling active learning method. We found that hyper-parameters for random sampling work well for other alternative approaches too. We scale all the features to have a mean zero and unit standard deviation.

Evaluation Metrics and Results

We evaluate the performance of the learned classifiers using the overall F-score on the evaluation set V, and report violations for parity-constrains in terms of demographic disparity and error rate balance (Table 1), as these account for different ends of the constrained spectrum of parity metrics. In order to set trade-off parameters (the convex combination for FAL and the constraints for fairlearn and PANDA), we first run FAL with several different trade-off parameters to find a value large enough that disparity matters but small enough that a non-zero F-score is possible. All results are with 0.6. We then observed the final disparity for FAL of 0.8 and set a constraint for PANDA and FAL of half of that: 0.4. This choice was made to ensure that FAL has an overall advantage over PANDA.

The main results are shown in Figure 2, where we consider performance for a fixed budget. Here, we first observe (unsurprisingly) that the baselines that do not take parity into account (Random Sampling and Entropy Sampling) do quite poorly (we do not plot margin-based sampling as it was dominated by Entropy sampling in all experiments). For example, while entropy sampling gets a very high F-score, it has quite poor disparity. Somewhat surprisingly, group-aware random sampling does worse in terms of disparity than even plain random sampling. FAL is able to achieve higher accuracy than random sampling, but, again, it's disparity is no better despite the fact that it explicitly optimizes for the trade-off. Finally, fairlearn and PANDA dominate in terms of the trade-off, with PANDA achieving higher accuracy, better error rate balance, but worth demographic disparity.

We also wish to consider the dynamic nature of these algorithms as they collect more data. In Figure 3, we plot budget versus f-score and disparity for a fixed parity constraint of 0.04. Unsurprisingly, we see that entropy sampling outperforms random sampling (in F-score), though they perform essentially the same for disparity. We also see a clear trade-off in FAL between F-score (goes up as the budget increases) and disparity (also goes up).

Here, we see that both fairlearn and PANDA are able to keep the disparity low (after an initial peak for PANDA). There is a generalization gap between PANDA's training disparity (which always exactly satisfies the 0.04 constraint) and its validation disparity, which is somewhat higher, as anticipated by concentration bounds on disparity like those of Agarwal et al. (2018). The initial peak in disparity (where it does not satisfy the constraint) for PANDA is not surprising: it is trained end-to-end to pick a good sample of 400 points; it is not optimized for smaller budgets. Similarly, in terms of F-score, PANDA achieves a very high initial F-score, essentially a zero-shot learning type effect. However, as it lowers the disparity, the F-score also drops slightly.

At test time, PANDA is the fastest of all the active learning algorithms we compare to from §4 with matching runtime performance to random sampling. This is because at test time we only need a single forward pass through the selection policy to select the B samples to label. Entropy sampling requires computing then entropy in every time step. Fairlearn is much slower as the learning reduction refits a mixture of experts model with different weights. Fair Active Learning is the slowest approach as it needs to compute the "expected fairness" that requires learning a new classifier for every data point in the pool of unlabeled data. For meta-training, learning the policy for PANDA converged after few hours of training.

Just Fscore & not accuracy is a bit weird - makes it challenging to place this result in context to other works.

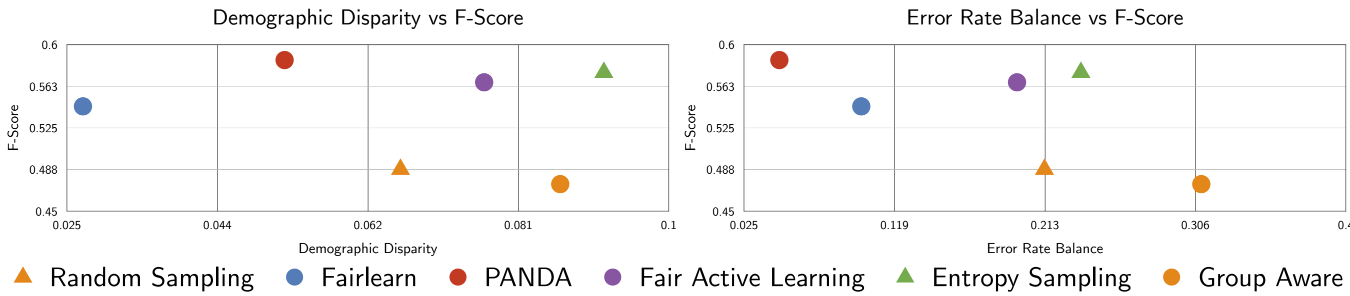

Figure 2: (Left) A scatterplot of demographic disparity versus F-score for a fixed budget $B = 400$, for PANDA and baseline approaches. (Right) A similar scatterplot for error rate balance versus F-score. In both cases, the upper-left is optimal behavior. Overall, we see that fairlearn and PANDA are the most competitive algorithms, with flipped behavior with respect to disparity on the two metrics. Dotted curves are algorithms unaware of parity/groups; solid lines are algorithms that are.

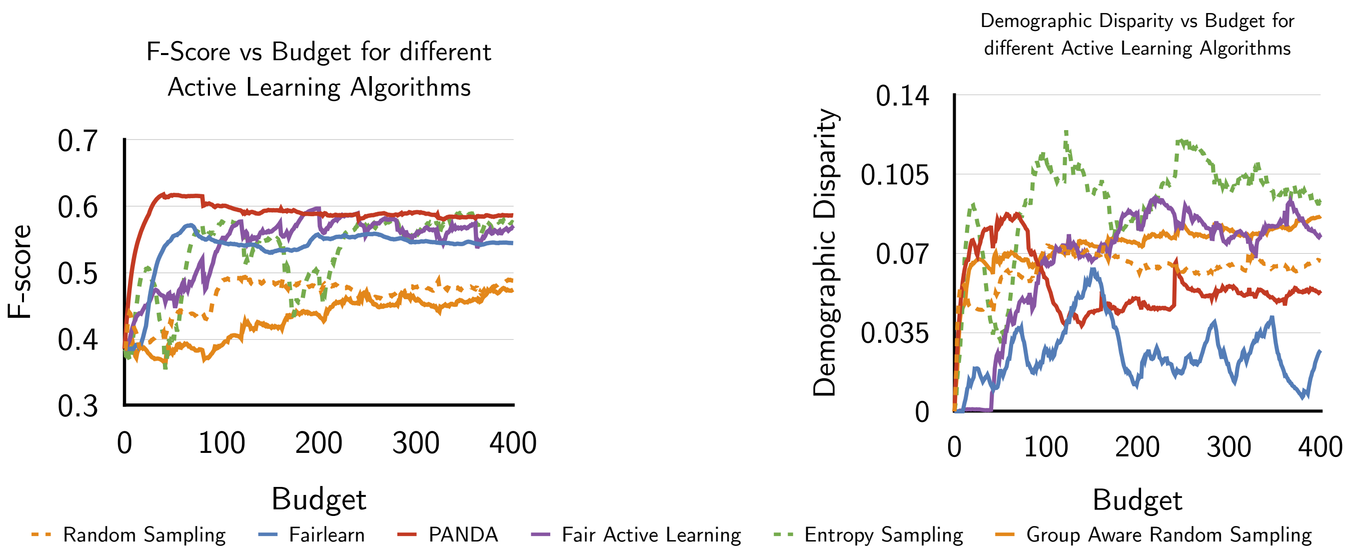

Figure 3: Learning curves for all algorithms, with (Left) budget (x-axis) versus F-score (y-axis) and (Right) budget (x-axis) versus demographic disparity (y-axis). The constraint value for fairlearn and PANDA is 0.04. Overall, we see that PANDA and fairlearn are able to (approximately) achieve the target parity, with PANDA achieving a higher F-score even than FAL (which has higher disparity).

why >0.04 ?

5 Discussion, Limitations and Conclusion

We presented PANDA, a meta-learning approach for learning to active learn under parity constraints, motivated by the desire to build an algorithm to mitigate unfairness in machine learning by collecting more data. We have seen that empirically PANDA is effective experimentally, even in a setting in which it essentially has to choose all 400 points to label at once, rather than one at a time. An obvious direction of future work is to incorporate features of the underlying classifier into the selection policy; the major challenge here is the computational expense of unrolling the corresponding computational graph. One major advantage of PANDA over all other alternatives is that in principle it does not need access to group information at test time. So long as it can be trained with group information available (for measuring disparities on the meta-test data), there is nothing in the algorithm that requires this information at test time. The only other setting in which this is possible is FAL with demographic disparity (precisely because demographic disparity does not need access to *labels*). Exploring this experimentally is a potential next step. Finally, there is the broader question of: how does one know what is the right intervention to mitigate disparities? Should we constrain our classifier? Should we collect more data? More features? Change the architecture? These are all important questions that are only beginning to be explored (Chen, Johansson, and Sontag 2018; Galhotra, Brun, and Meliou 2017; Udeshi, Arora, and Chattopadhyay 2018; Angell et al. 2018).

Standard concentration bounds on disparity like those of (Agarwal et al. 2018) hold for PANDA. In the sense that if a given disparity is achieved at training time, then PANDA guarantees a bound on the generalization error for the validation disparity. However, what one would like to show is that after sampling B labeled examples using PANDA, we can guarantee improvement in fairness parity; this theoretical analysis remains a future work item.

References

Abdollahi, B.; and Nasraoui, O. 2018. Transparency in fair machine learning: The case of explainable recommender systems. In *Human and Machine Learning*, 21–35. Springer.

Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453* .

Anahideh, H.; and Asudeh, A. 2020. Fair Active Learning. *arXiv preprint arXiv:2001.01796* .

Angell, R.; Johnson, B.; Brun, Y.; and Meliou, A. 2018. Themis: Automatically testing software for discrimination. In *Joint Meeting on European Software Engineering*.

Angluin, D. 1988. Queries and Concept Learning. *Mach. Learn.* 2(4): 319–342. ISSN 0885-6125. doi:10.1023/A:1022821128753. URL `https://doi.org/10.1023/A:1022821128753`.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine bias. ProPublica. *See https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing* .

Bachman, P.; Sordoni, A.; and Trischler, A. 2017. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 301–310. JMLR. org.

Cabrera, Á. A.; Epperson, W.; Hohman, F.; Kahng, M.; Morgenstern, J.; and Chau, D. H. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. *arXiv preprint arXiv:1904.05419* .

Chen, I.; Johansson, F. D.; and Sontag, D. 2018. Why is my classifier discriminatory? In *Advances in Neural Information Processing Systems*, 3539–3550.

Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5(2): 153–163.

Cohn, D.; Atlas, L.; and Ladner, R. 1994. Improving generalization with active learning. *Machine learning* 15(2): 201–221.

Combettes, P. L.; and Wajs, V. R. 2005. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation* 4(4): 1168–1200.

Crawford, K.; and Calo, R. 2016. There is a blind spot in AI research. *Nature* 538(7625): 311–313.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* .

Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; and Kaiser, Ł. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL `http://archive.ics.uci.edu/ml`.

Fang, M.; Li, Y.; and Cohn, T. 2017. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383* .

Fedorov, V. V. 2013. *Theory of optimal experiments*. Elsevier.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and Removing Disparate Impact.

Friedman, B.; and Nissenbaum, H. 1996. Bias in computer systems. *ACM Transactions on Information Systems (TOIS)* 14(3): 330–347.

Galhotra, S.; Brun, Y.; and Meliou, A. 2017. Fairness testing: Testing software for discrimination. In *Joint Meeting on Foundations of Software Engineering*.

Giang, V. 2018. The potential hidden bias in automated hiring systems. *Fast Company* .

Goldstein, T.; Studer, C.; and Baraniuk, R. 2014. A field guide to forward-backward splitting with a FASTA implementation. *arXiv preprint arXiv:1411.3406* .

Gumbel, E. J. 1948. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, 3315–3323.

Holstein, K.; Wortman Vaughan, J.; Daumé III, H.; Dudik, M.; and Wallach, H. 2019. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–16.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Jiang, J.; and Ip, H. H.-S. 2008. Active learning for the prediction of phosphorylation sites. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 3158–3165. IEEE.

Kallus, N.; and Zhou, A. 2018. Residual unfairness in fair machine learning from prejudiced data. *arXiv preprint arXiv:1806.02887* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kleinberg, J.; Mullainathan, S.; and Raghavan, M. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* .

Konyushkova, K.; Sznitman, R.; and Fua, P. 2017. Learning active learning from data. In *Advances in Neural Information Processing Systems*, 4225–4235.

Kool, W.; Van Hoof, H.; and Welling, M. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. *arXiv preprint arXiv:1903.06059* .

Lions, P.-L.; and Mercier, B. 1979. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis* 16(6): 964–979.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* .

Narayanan, A. 2018. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Rai, P.; Saha, A.; Daumé III, H.; and Venkatasubramanian, S. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 27–32. Association for Computational Linguistics.

Roth, D.; and Small, K. 2006. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, 413–424. Springer.

Settles, B. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Shannon, C. E. 1948. A note on the concept of entropy. *Bell System Tech. J* 27(3): 379–423.

Shi, X.; Fan, W.; and Ren, J. 2008. Actively transfer domain knowledge. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 342–357. Springer.

Srivastava, M.; Heidari, H.; and Krause, A. 2019. Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2459–2468.

Udeshi, S.; Arora, P.; and Chattopadhyay, S. 2018. Automated directed fairness testing. In *International Conference on Automated Software Engineering*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veale, M.; and Binns, R. 2017. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society* 4(2): 2053951717743530.

Vieira, T. 2014. Gumbel-max trick and weighted reservoir sapling. https://timvieira.github.io/blog/post/2014/08/01/gumbel-max-trick-and-weightedreservoir-sampling/.

Wachter-Boettcher, S. 2017. AI recruiting tools do not eliminate bias. *Time Magazine* .

Wang, X.; Huang, T.-K.; and Schneider, J. 2014. Active transfer learning under model shift. In *International Conference on Machine Learning*, 1305–1313.

Wightman, L. 1998. LSAC National Longitudinal Bar Passage Study.

Wilson, B.; Hoffman, J.; and Morgenstern, J. 2019. Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097* .

Zarsky, T. 2016. The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. *Science, Technology, & Human Values* 41(1): 118–132.

Broader Impact

The motivation of this work is precisely to have positive broader impacts, by giving machine learning practitioners who care about fairness in machine learning another tool in their toolbox to build models with fewer disparities. Our primary target stakeholder population is such machine learning practitioners and data scientists. Secondarily, as that primary stakeholder population builds and deploys algorithms, those who are impacted by those algorithms through direct or indirect use will, we hope, suffer fewer disparities as a result.

There are several risks. The first is a false sense of security. For instance, we do not prove formally that this approach is guaranteed to work in all cases, and our empirical results are limited to a small number of tests over a single dataset. On the positive side, Agarwal et al. (2018) prove a generalization bound for disparity that applies to our algorithm (as well as any other algorithm); thus, so long as practitioners properly test the resulting disparities of their models, they can consult these generalization bounds to get estimates of worst case behavior.

A second risk is around, if deployed, how the new data is collected. We have seen news stories recently around unethical practices for data collection. Any additional labeling that is performed as a result of running this or similar algorithms should be done consistent with standard ethical guidelines for data collection.

Overall, while there are real concerns about how this technology might be deployed, our hope is that the positive impacts outweigh the negatives, specifically because standard best-use practices should mitigate most of the risks.