Online Learning with an Unknown Fairness Metric

Stephen Gillen[*] Christopher Jung[†] Michael Kearns[‡] Aaron Roth[§]

September 17, 2018

Abstract

We consider the problem of online learning in the linear contextual bandits setting, but in which there are also strong *individual fairness* constraints governed by an unknown similarity metric. These constraints demand that we select similar actions or individuals with approximately equal probability [Dwork et al., 2012], which may be at odds with optimizing reward, thus modeling settings where profit and social policy are in tension. We assume we learn about an unknown Mahalanobis similarity metric from only weak feedback that identifies fairness violations, but does not quantify their extent. This is intended to represent the interventions of a regulator who "knows unfairness when he sees it" but nevertheless cannot enunciate a quantitative fairness metric over individuals. Our main result is an algorithm in the adversarial context setting that has a number of fairness violations that depends only logarithmically on T, while obtaining an optimal $O(\sqrt{T})$ regret bound to the best fair policy.

[*]Department of Mathematics, University of Pennsylvania.

[†]Department of Computer and Information Sciences, University of Pennsylvania. Supported in part by a grant from the Quattrone Center for the Fair Administration of Justice.

[‡]Department of Computer and Information Sciences, University of Pennsylvania.

[§]Department of Computer and Information Sciences, University of Pennsylvania. Supported in part by grants from the DARPA Brandeis project, the Sloan Foundation, and NSF grants CNS-1513694 and CNS-1253345.

1 Introduction

The last several years have seen an explosion of work studying the problem of fairness in machine learning. Yet there remains little agreement about what "fairness" should mean in different contexts. In broad strokes, the literature can be divided into two families of fairness definitions: those aiming at *group* fairness, and those aiming at *individual* fairness.

Group fairness definitions are aggegrate in nature: they partition individuals into some collection of *protected groups* (say by race or gender), specify some statistic of interest (say, positive classification rate or false positive rate), and then require that a learning algorithm equalize this quantity across the protected groups. On the other hand, individual fairness definitions ask for some constraint that binds on the individual level, rather than only over averages of people. Often, these constraints have the semantics that "similar people should be treated similarly" Dwork et al. [2012].

Individual fairness definitions have substantially stronger semantics and demands than group definitions of fairness. For example, Dwork et al. [2012] lay out a compendium of ways in which group fairness definitions are unsatisfying. Yet despite these weaknesses, group fairness definitions are by far the most prevalent in the literature (see e.g. Kamiran and Calders [2012], Hajian and Domingo-Ferrer [2013], Kleinberg et al. [2017], Hardt et al. [2016], Friedler et al. [2016], Zafar et al. [2017], Chouldechova [2017] and Berk et al. [2017] for a survey). This is in large part because notions of individual fairness require making stronger assumptions on the setting under consideration. In particular, the definition from Dwork et al. [2012] requires that the algorithm designer know a "task-specific fairness metric."

Learning problems over individuals are also often implicitly accompanied by some notion of *merit*, embedded in the objective function of the learning problem. For example, in a lending setting we might posit that each loan applicant is either "creditworthy" and will repay a loan, or is not creditworthy and will default — which is what we are trying to predict. Joseph et al. [2016a] take the approach that this measure of merit — already present in the model, although initially unknown to the learner — can be taken to be the similarity metric in the definition of Dwork et al. [2012], requiring informally that creditworthy individuals have at least the same probability of being accepted for loans as defaulting individuals. (The implicit and coarse fairness metric here assigns distance zero between pairs of creditworthy individuals and pairs of defaulting individuals, and some non-zero distance between a creditworthy and a defaulting individual.) This resolves the problem of how one should discover the "fairness metric", but results in a notion of fairness that is necessarily aligned with the notion of "merit" (creditworthiness) that we are trying to predict.

However, there are many settings in which the notion of merit we wish to predict may be different or even at odds with the notion of fairness we would like to enforce. For example, notions of fairness aimed at rectifying societal inequities that result from historical discrimination can aim to favor the disadvantaged population (say, in college admissions), even if the performance of the admitted members of that population can be expected to be lower than that of the advantaged population. Similarly, we might desire a fairness metric incorporating only those attributes that individuals can change in principle (and thus excluding ones like race, age and gender), and that further expresses what are and are not meaningful differences between individuals, outside the context of any particular prediction problem. These kinds of fairness desiderata can still be expressed as an instantiation of the definition from Dwork et al. [2012], but with a task-specific

fairness metric separate from the notion of merit we are trying to predict.

In this paper, we revisit the individual fairness definition from Dwork et al. [2012]. This definition requires that pairs of individuals who are close in the fairness metric must be treated "similarly" (e.g. in an allocation problem such as lending, served with similar probability). We investigate the extent to which it is possible to satisfy this fairness constraint while simultaneously solving an online learning problem, when the underlying fairness metric is Mahalanobis but *not* known to the learning algorithm, and may also be in tension with the learning problem. One conceptual problem with metric-based definitions, that we seek to address, is that it may be difficult for anyone to actually precisely express a quantitative metric over individuals — but they nevertheless might "know unfairness when they see it." We therefore assume that the algorithm has access to an oracle that knows intuitively what it means to be fair, but cannot explicitly enunciate the fairness metric. Instead, given observed actions, the oracle can specify whether they were fair or not, and the goal is to obtain low regret in the online learning problem — measured with respect to the best *fair* policy — while also limiting violations of individual fairness during the learning process.

1.1 Our Results and Techniques

We study the standard linear contextual bandit setting. In rounds $t = 1, \ldots, T$, a learner observes arbitrary and possibly adversarially selected d-dimensional contexts, each corresponding to one of k actions. The reward for each action is (in expectation) an unknown linear function of the contexts. The learner seeks to minimize its regret.

The learner also wishes to satisfy *fairness constraints*, defined with respect to an unknown distance function defined over contexts. The constraint requires that the difference between the probabilities that any two actions are taken is bounded by the distance between their contexts. The learner has no initial knowledge of the distance function. Instead, after the learner makes its decisions according to some probability distribution π^t at round t, it receives feedback specifying for which pairs of contexts the fairness constraint was violated. Our goal in designing a learner is to simultaneously guarantee near-optimal regret in the contextual bandit problem (with respect to the best *fair* policy), while violating the fairness constraints as infrequently as possible. Our main result is a computationally efficient algorithm that guarantees this for a large class of distance functions known as *Mahalanobis distances* (these can be expressed as $d(x_1, x_2) = \|Ax_1 - Ax_2\|_2$ for some matrix A).

Theorem (Informal): There is a computationally efficient learning algorithm $\boldsymbol{L}$ in our setting that guarantees that for any Mahalanobis distance, any time horizon T, and any error tolerance ϵ:

1. (Learning) With high probability, $\boldsymbol{L}$ obtains regret $\tilde{O}\left(k^2 d^2 \log(T) + d\sqrt{T}\right)$ to the best fair policy (See Theorem 3 for a precise statement.)

2. (Fairness) With probability 1, $\boldsymbol{L}$ violates the unknown fairness constraints by more than ϵ on at most $O\left(k^2 d^2 \log(d/\epsilon)\right)$ many rounds. (Theorem 4.)

We note that the quoted regret bound requires setting $\epsilon = O(1/T)$, and so this implies a number of fairness violations of magnitude more than $1/T$ that is bounded by a function growing logarithmically in T. Other tradeoffs between regret and fairness violations are possible.

These two goals: of obtaining low regret, and violating the unknown constraint a small number of times — are seemingly in tension. A standard technique for obtaining a mistake bound with

respect to fairness violations would be to play a "halving algorithm", which would always act as if the unknown metric is at the center of the current version space (the set of metrics consistent with the feedback observed thus far) — so that mistakes necessarily remove a non-trivial fraction of the version space, making progress. On the other hand, a standard technique for obtaining a diminishing regret bound is to play "optimistically" – i.e. to act as if the unknown metric is the point in the version space that would allow for the largest possible reward. But "optimistic" points are necessarily at the boundary of the version space, and when they are falsified, the corresponding mistakes do not necessarily reduce the version space by a constant fraction.

We prove our theorem in two steps. First, in Section 3, we consider the simpler problem in which the linear objective of the contextual bandit problem is known, and the distance function is all that is unknown. In this simpler case, we show how to obtain a bound on the number of fairness violations using a linear-programming based reduction to a recent algorithm which has a mistake bound for learning a linear function with a particularly weak form of feedback Lobel et al. [2017]. A complication is that our algorithm does not receive all of the feedback that the algorithm of Lobel et al. [2017] expects. We need to use the structure of our linear program to argue that this is ok. Then, in Section 4, we give our algorithm for the complete problem, using large portions of the machinery we develop in Section 3.

We note that in a non-adversarial setting, in which contexts are drawn from a distribution, the algorithm of Lobel et al. [2017] could be more simply applied along with standard techniques for contextual bandit learning to give an explore-then-exploit style algorithm. This algorithm would obtain bounded (but suboptimal) regret, and a number of fairness violations that grows as a root of T. The principal advantages of our approach are that we are able to give a number of fairness violations that has only *logarithmic* dependence on T, while tolerating contexts that are chosen adversarially, all while obtaining an optimal $O(\sqrt{T})$ regret bound to the best fair policy.

1.2 Additional Related Work

There are two papers, written concurrently to ours, that tackle orthogonal issues in metric-fair learning. Rothblum and Yona [2018] consider the problem of *generalization* when performing learning subject to a known metric constraint. They show that it is possible to prove relaxed PAC-style generalization bounds without any assumptions on the metric, and that for worst-case metrics, learning subject to a metric constraint can be computationally hard, even when the unconstrained learning problem is easy. In contrast, our work focuses on online learning with an *unknown* metric constraint. Our results imply similar generalization properties via standard online-to-offline reductions, but only for the class of metrics we study. Kim et al. [2018] considers a group-fairness like relaxation of metric-fairness, asking that on average, individuals in pre-specified groups are classified with probabilities proportional to the average distance between individuals in those groups. They show how to learn such classifiers in the offline setting, given access to an oracle which can evaluate the distance between two individuals according to the metric (allowing for unbiased noise). The similarity to our work is that we also consider access to the fairness metric via an oracle, but our oracle is substantially weaker, and does not provide numeric valued output.

There are also several papers in the algorithmic fairness literature that are thematically related to ours, in that they both aim to bridge the gap between group notions of fairness (which can be semantically unsatisfying) and individual notions of fairness (which require very strong assumptions). Zemel et al. [2013] attempt to automatically learn a representation for the data in a

batch learning problem (and hence, implicitly, a similarity metric) that causes a classifier to label an equal proportion of two protected groups as positive. They provide a heuristic approach and an experimental evaluation. Two recent papers (Kearns et al. [2017] and Hébert-Johnson et al. [2017]) take the approach of asking for a group notion of fairness, but over exponentially many implicitly defined protected groups, thus mitigating what Kearns et al. [2017] call the "fairness gerrymandering" problem, which is one of the principal weaknesses of group fairness definitions. Both papers give polynomial time reductions which yield efficient algorithms whenever a corresponding agnostic learning problem is solvable. In contrast, in this paper, we take a different approach: we attempt to directly satisfy the original definition of individual fairness from Dwork et al. [2012], but with substantially less information about the underlying similarity metric.

Starting with Joseph et al. [2016a], several papers have studied notions of fairness in classic and contextual bandit problems. Joseph et al. [2016a] study a notion of "meritocratic" fairness in the contextual bandit setting, and prove upper and lower bounds on the regret achievable by algorithms that must be "fair" at every round. This can be viewed as a variant of the Dwork et al. [2012] notion of fairness, in which the expected reward of each action is used to define the "fairness metric". The algorithm does not originally know this metric, but must discover it through experimentation. Joseph et al. [2016b] extend the work of Joseph et al. [2016a] to the setting in which the algorithm is faced with a continuum of options at each time step, and give improved bounds for the *linear* contextual bandit case. Jabbari et al. [2017] extend this line of work to the reinforcement learning setting in which the actions of the algorithm can impact its environment. Finally, Liu et al. [2017] consider a notion of fairness based on calibration in the simple stochastic bandit setting.

There is a large literature that focuses on learning Mahalanobis distances — see Kulis et al. [2013] for a survey. In this literature, the closest paper to our work focuses on *online* learning of Mahalanobis distances (Jain et al. [2009]). However, this result is in a very different setting from the one we consider here. In Jain et al. [2009], the algorithm is repeatedly given pairs of points, and needs to predict their distance. It then learns their true distance, and aims to minimize its squared loss. In contrast, in our paper, the main objective of the learning algorithm is orthogonal to the metric learning problem — i.e. to minimize regret in the linear contextual bandit problem, but while simultaneously learning and obeying a fairness constraint, and only from weak feedback noting violations of fairness.

2 Model and Preliminaries

2.1 Linear Contextual Bandits

We study algorithms that operate in the *linear contextual bandits* setting. A linear contextual bandit problem is parameterized by an unknown vector of linear coefficients $\theta \in \mathbb{R}^d$, with $\|\theta\|_2 \leq 1$. Algorithms in this setting operate in *rounds* $t = 1, \ldots, T$. In each round t, an algorithm $\boldsymbol{L}$ observes k *contexts* $x_1^t, \ldots, x_k^t \in \mathbb{R}^d$, scaled such that $\|x_i^t\|_2 \leq 1$. We write $x^t = (x_1^t, \ldots, x_k^t)$ to denote the entire set of contexts observed at round t. After observing the contexts, the algorithm chooses an action i^t. After choosing an action, the algorithm obtains some stochastic *reward* $r_{i^t}^t$ such that $r_{i^t}^t$ is subgaussian[1] and $\mathbb{E}[r_{i^t}^t] = \langle x_{i^t}^t, \theta \rangle$. The algorithm does not observe the reward for the actions not chosen. When the action i^t is clear from context, and write r^t instead of $r_{i^t}^t$.

[1] A random variable X with $\mu = \mathbb{E}[X]$ is sub-gaussian, if for all $t \in \mathbb{R}$, $\mathbb{E}[e^{t(X-\mu)}] \leq e^{\frac{t^2}{2}}$.

Remark 1. *For simplicity, we consider algorithms that select only a* single *action at every round. However, this assumption is not necessary. In the appendix, we show how our results extend to the case in which the algorithm can choose any number of actions at each round. This assumption is sometimes more natural: for example, in a lending scenario, a bank may wish to make loans to as many individuals as will be profitable, without a budget constraint.*

In this paper, we will be discussing algorithms $\boldsymbol{L}$ that are necessarily randomized. To formalize this, we denote a history including everything observed by the algorithm up through but not including round t as $h^t = ((x^1, i^1, r^1), \ldots, (x^{t-1}, i^{t-1}, r^{t-1}))$ The space of such histories is denoted by $\mathcal{H}^t = (\mathbb{R}^{d\times k} \times [k] \times \mathbb{R})^{t-1}$. An algorithm $\boldsymbol{L}$ is defined by a sequence of functions $f^1, \ldots, f^T$ each mapping histories and observed contexts to probability distributions over actions:

$$f^t : \mathcal{H}^t \times \mathbb{R}^{d\times k} \to \Delta[k].$$

We write π^t to denote the probability distribution over actions that $\boldsymbol{L}$ plays at round t: $\pi^t = f^t(h^t, x^t)$. We view π^t as a vector over $[0,1]^k$, and so π^t_i denotes the probability that $\boldsymbol{L}$ plays action i at round t. We denote the expected reward of the algorithm at day t as $\mathbb{E}[r^t] = \mathbb{E}_{i\sim\pi^t}[r^t_i]$. It will sometimes also be useful to refer to the vector of expected rewards across all actions on day t. We denote it as

$$\bar{r}^t = (\langle x^t_1, \theta\rangle, \ldots, \langle x^t_k, \theta\rangle).$$

Note that this vector is of course unknown to the algorithm.

2.2 Fairness Constraints and Feedback

We study algorithms that are constrained to behave *fairly* in some manner. We adapt the definition of fairness from Dwork et al. [2012] that asserts, informally, that "similar individuals should be treated similarly". We imagine that the decisions that our contextual bandit algorithm $\boldsymbol{L}$ makes correspond to individuals, and that the contexts x^t_i correspond to features pertaining to individuals. We adopt the following (specialization of) the fairness definition from Dwork et al, which is parameterized by a distance function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

Definition 1 (Dwork et al. [2012])**.** *Algorithm $\boldsymbol{L}$ is Lipschitz-fair on round t with respect to distance function d if for all pairs of individuals i, j:*

$$|\pi^t_i - \pi^t_j| \leq d(x^t_i, x^t_j).$$

For brevity, we will often just say that the algorithm is fair *at round t, with the understanding that we are always talking about this one particular kind of fairness.*

Remark 2. *Note that this definition requires a fairness constraint that binds between individuals at a single round t, but not between rounds t. This is for several reasons. First, at a philosophical level, we want our algorithms to be able to improve with time, without being bound by choices they made long ago before they had any information about the fairness metric. At a (related) technical level, it is easy to construct lower bound instances that certify that it is impossible to simultaneously guarantee that an algorithm has diminishing regret to the best fair policy, while violating fairness constraints (now defined as binding across rounds) a sublinear number of times.*

One of the main difficulties in working with Lipschitz fairness (as discussed in Dwork et al. [2012]) is that the distance function d plays a central role, but it is not clear how it should be specified. In this paper, we concern ourselves with learning d from feedback. In particular, algorithms $\boldsymbol{L}$ will have access to a *fairness oracle*.

Informally, the fairness oracle will take as input: 1) the set of choices available to $\boldsymbol{L}$ at each round t, and 2) the probability distribution π^t that $\boldsymbol{L}$ uses to make its choices at round t, and returns the set of all pairs of individuals for which $\boldsymbol{L}$ violates the fairness constraint.

Definition 2 (Fairness Oracle)**.** *Given a distance function d, a fairness oracle $\boldsymbol{O}_d$ is a function $\boldsymbol{O}_d : \mathbb{R}^{d\times k} \times \Delta[k] \to 2^{[k]\times[k]}$ defined such that:*

$$\boldsymbol{O}_d(x^t, \pi^t) = \{(i,j) : |\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t)\}$$

Formally, algorithms $\boldsymbol{L}$ in our setting will operate in the following environment:

Definition 3.

1. *An adversary fixes a linear reward function $\theta \in \mathbb{R}^d$ with $\|\theta\| \leq 1$ and a distance function d. $\boldsymbol{L}$ is given access to the fairness oracle $\boldsymbol{O}_d$.*
2. *In rounds $t = 1$ to T:*
   - *(a) The adversary chooses contexts $x^t \in \mathbb{R}^{d\times k}$ with $\|x_i^t\| \leq 1$ and gives them to $\boldsymbol{L}$.*
   - *(b) $\boldsymbol{L}$ chooses a probability distribution π^t over actions, and chooses action $i^t \sim \pi^t$.*
   - *(c) $\boldsymbol{L}$ receives reward $r_{i^t}^t$ and observes feedback $\boldsymbol{O}_d(\pi^t)$ from the fairness oracle.*

Because of the power of the adversary in this setting, we cannot expect algorithms that can avoid arbitrarily small violations of the fairness constraint. Instead, we will aim to limit *significant* violations.

Definition 4. *Algorithm $\boldsymbol{L}$ is ϵ-unfair on pair (i,j) at round t with respect to distance function d if*

$$|\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t) + \epsilon.$$

Given a sequence of contexts and a history h^t (which fixes the distribution on actions at day t) We write

$$\textbf{Unfair}(\boldsymbol{L}, \epsilon, h^t) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \mathbb{1}(|\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t) + \epsilon)$$

to denote the number of pairs on which $\boldsymbol{L}$ is ϵ-unfair at round t.

Given a distance function d and a history h^{T+1}, the *ϵ-fairness loss* of an algorithm $\boldsymbol{L}$ is the total number of pairs on which it is ϵ-unfair:

$$\textbf{FairnessLoss}(\boldsymbol{L}, h^{T+1}, \epsilon) = \sum_{t=1}^{T} \textbf{Unfair}(\boldsymbol{L}, \epsilon, h^t)$$

For a shorthand, we'll write $\textbf{FairnessLoss}(\boldsymbol{L}, T, \epsilon)$.

We will aim to design algorithms $\boldsymbol{L}$ that guarantee that their fairness loss is bounded with probability 1 in the worst case over the instance: i.e. in the worst case over both θ and $x^1, \ldots, x^T$, and in the worst case over the distance function d (within some allowable class of distance functions – see Section 2.4).

2.3 Regret to the Best Fair Policy

In addition to minimizing fairness loss, we wish to design algorithms that exhibit diminishing *regret* to the best *fair* policy. We first define a linear program that we will make use of throughout the paper. Given a vector $a \in \mathbb{R}^d$ and a vector $c \in \mathbb{R}^{k^2}$, we denote by $LP(a,c)$ the following linear program:

$$\begin{aligned} \underset{\pi=\{p_1,\ldots,p_k\}}{\text{maximize}} \quad & \sum_{i=1}^{k} p_i a_i \\ \text{subject to} \quad & |p_i - p_j| \leq c_{i,j}, \forall (i,j) \\ & \sum_{i=1}^{k} p_i \leq 1 \end{aligned}$$

We write $\pi(a,c) \in \Delta[k]$ to denote an optimal solution to $LP(a,c)$. Given a set of contexts x^t, recall that $\bar{r}^t$ is the vector representing the expected reward corresponding to each context (according to the true, unknown linear reward function θ). Similarly, we write $\bar{d}^t$ to denote the vector representing the set of distances between each pair of contexts i,j (according to the true, unknown distance function d): $\bar{d}^t_{i,j} = d(x^t_i, x^t_j)$.

Observe that $\pi(\bar{r}^t, \bar{d}^t)$ corresponds to the distribution over actions that maximizes expected reward at round t, subject to satisfying the fairness constraints — i.e. the distribution that an optimal player, with advance knowledge of θ would play, if he were not allowed to violate the fairness constraints at all. This is the benchmark with respect to which we define regret:

Definition 5. *Given an algorithm $\boldsymbol{L}$ $(f_1,\ldots,f_T)$, a distance function d, a linear parameter vector θ, and a history h^{T+1} (which includes a set of contexts $x^1,\ldots,x^T$), its regret is defined to be:*

$$\textbf{Regret}(\boldsymbol{L},\theta,d,h^{T+1}) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\pi(\bar{r}^t,\bar{d}^t)}[\bar{r}^t_i] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim f^t(h^t,x^t)}[\bar{r}^t_i]$$

For shorthand, we'll write $\textbf{Regret}(\boldsymbol{L},T)$.

Our goal will be to design algorithms for which we can bound regret with high probability over the randomness of h^{T+1} [2] in the worst case over θ, d, and $(x^1,\ldots,x^T)$.

2.4 Mahalanobis Distance

In this paper, we will restrict our attention to a special family of distance functions which are parameterized by a matrix A:

Definition 6 (Mahalanobis distances)**.** *A function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a Mahalanobis distance function if there exists a matrix A such that for all $x_1, x_2 \in \mathbb{R}^d$:*

$$d(x_1,x_2) = \|Ax_1 - Ax_2\|_2$$

where $\|\cdot\|_2$ denotes Euclidean distance. Note that if A is not full rank, then this does not define a metric — but we will allow this case (and be able to handle it in our algorithmic results).

[2] We assume that h^{T+1} is generated by algorithm A, meaning randomness only comes from the stochastic reward and the way in which each arm is selected according to the probability distribution calculated by the algorithm. We don't assume any distributional assumption over $x^1,\ldots,x^T$

Mahalanobis distances will be convenient for us to work with, because *squared* Mahalanobis distances can be expressed as follows:

$$\begin{aligned} d(x_1, x_2)^2 &= \|Ax_1 - Ax_2\|_2^2 \\ &= \langle A(x_1 - x_2), A(x_1 - x_2)\rangle \\ &= (x_1 - x_2)^\top A^\top A(x_1 - x_2) \\ &= \sum_{i,j=1}^{d} G_{i,j}(x_1 - x_2)_i(x_1 - x_2)_j \end{aligned}$$

where $G = A^\top A$. Observe that when x_1 and x_2 are fixed, this is a linear function in the entries of the matrix G. We will use this property to reason about *learning* G, and thereby learning d.

3 Warmup: The Known Objective Case

In this section, we consider an easier case of the problem in which the linear objective function θ is known to the algorithm, and the distance function d is all that is unknown. In this case, we show via a reduction to an online learning algorithm of Lobel et al. [2017], how to simultaneously obtain a logarithmic regret bound and a logarithmic (in T) number of fairness violations. The analysis we do here will be useful when we solve the full version of our problem (in which θ is unknown) in Section 4.

3.1 Outline of the Solution

Recall that since we know θ, at every round t after seeing the contexts, we know the vector of expected rewards $\bar{r}^t$ that we would obtain for selecting each action. Our algorithm will play at each round t the distribution $\pi(\bar{r}^t, \hat{d}^t)$ that results from solving the linear program $LP(\bar{r}^t, \hat{d}^t)$, where $\hat{d}^t$ is a "guess" for the pairwise distances between each context $\bar{d}^t$. (Recall that the optimal distribution to play at each round is $\pi(\bar{r}^t, \bar{d}^t)$.)

The main engine of our reduction is an efficient online learning algorithm for linear functions recently given by Lobel et al. [2017] which is further described in Section 3.2. Their algorithm, which we refer to as **DistanceEstimator**, works in the following setting. There is an unknown vector of linear parameters $\alpha \in \mathbb{R}^m$. In rounds t, the algorithm observes a vector of features $u^t \in \mathbb{R}^m$, and produces a prediction $g^t \in \mathbb{R}$ for the value $\langle \alpha, u^t \rangle$. After it makes its prediction, the algorithm learns whether its guess was *too large* or not, but does not learn anything else about the value of $\langle \alpha, u^t \rangle$. The guarantee of the algorithm is that the number of rounds in which its prediction is off by more than ϵ is bounded by $O(m \log(m/\epsilon))$[3].

Our strategy will be to instantiate $\binom{k}{2}$ copies of this distance estimator — one for each pair of actions — to produce guesses $(\hat{d}^t_{i,j})^2$ intended to approximate the *squared* pairwise distances $d(x^t_i, x^t_j)^2$. From this we derive estimates $\hat{d}^t_{i,j}$ of the pairwise distances $d(x^t_i, x^t_j)$. Note that this is a linear estimation problem for any Mahalanobis distance, because by our observation in Section

[3] If the algorithm also learned whether or not its guess was in error by more than ϵ at each round, variants of the classical halving algorithm could obtain this guarantee. But the algorithm does not receive this feedback, which is why the more sophisticated algorithm of Lobel et al. [2017] is needed.

2.4, a squared Mahalanobis distance can be written as a linear function of the $m = d^2$ unknown entries of the matrix $G = A^\top A$ which defines the Mahalanobis distance.

The complication is that the **DistanceEstimator** algorithms expect feedback at every round, which we cannot always provide. This is because the fairness oracle $\boldsymbol{O}_d$ provides feedback about the distribution $\pi(\bar{r}^t, \hat{d}^t)$ used by the algorithm, *not* directly about the guesses $\hat{d}^t$. These are not the same, because not all of the constraints in the linear program $LP(\bar{r}^t, \hat{d}^t)$ are necessarily tight — it may be that $|\pi(\bar{r}^t, \hat{d}^t)_i - \pi(\bar{r}^t, \hat{d}^t)_j| < \hat{d}^t_{i,j}$. For any copy of **DistanceEstimator** that does not receive feedback, we can simply "roll back" its state and continue to the next round. But we need to argue that we make progress — that whenever we are ϵ-unfair, or whenever we experience large per-round regret, then there is at least one copy of **DistanceEstimator** that we can give feedback to such that the corresponding copy of **DistanceEstimator** has made a large prediction error, and we can thus charge either our fairness loss or our regret to the mistake bound of that copy of **DistanceEstimator**.

As we show, there are three relevant cases.

1. In any round in which we are ϵ-unfair for some pair of contexts x^t_i and x^t_j, then it must be that $\hat{d}^t_{i,j} \geq d(x^t_i, x^t_j) + \epsilon$, and so we can always update the (i,j)th copy of **DistanceEstimator** and charge our fairness loss to its mistake bound. We formalize this in Lemma 1.

2. For any pair of arms (i,j) such that we have not violated the fairness constraint, *and* the (i,j)th constraint in the linear program is tight, we can provide feedback to the (i,j)th copy of **DistanceEstimator** (its guess was not too large). There are two cases. Although the algorithm never knows which case it is in, we handle each case separately in the analysis.

   (a) For every constraint (i,j) in $LP(\bar{r}^t, \hat{d}^t)$ that is *tight* in the optimal solution, $|\hat{d}^t_{i,j} - d(x^t_i, x^t_j)| \leq \epsilon$. In this case, we show that our algorithm does not incur very much per round regret. We formalize this in Lemma 4.

   (b) Otherwise, there is a tight constraint (i,j) such that $|\hat{d}^t_{i,j} - d(x^t_i, x^t_j)| > \epsilon$. In this case, we may incur high per-round regret — but we can charge such rounds to the mistake bound of the (i,j)th copy of **DistanceEstimator** using Lemma 1.

3.2 The Distance Estimator

First, we fix some notation for the **DistanceEstimator** algorithm. We write **DistanceEstimator**(ϵ) to instantiate a copy of **DistanceEstimator** with a mistake bound for ϵ-misestimations. The mistake bound we state for **DistanceEstimator** is predicated on the assumption that the norm of the unknown linear parameter vector $\alpha \in \mathbb{R}^m$ is bounded by $\|\alpha\| \leq B_1$, and the norms of the arriving vectors $u^t \in \mathbb{R}^m$ are bounded by $\|u^t\| \leq B_2$. Given an instantiation of **DistanceEstimator** and a new vector u^t for which we would like a prediction, we write: $g^t =$ **DistanceEstimator**$.guess(u^t)$ for its guess of the value of $\langle \alpha, u^t \rangle$. We use the following notation to refer to the feedback we provide to **DistanceEstimator**: If $g^t > \langle \alpha, u^t \rangle$ and we provide feedback, we write **DistanceEstimator**$.feedback(\top)$. Otherwise, if $g^t \leq \langle \alpha, u^t \rangle$ and we give feedback, we write **DistanceEstimator**$.feedback(\bot)$. In some rounds, we may be unable to provide the feedback that **DistanceEstimator** is expecting: in these rounds, we simply "roll-back" its internal state. We can do this because the mistake bound for **DistanceEstimator** holds for *every* sequence of arriving vectors u^t. If we give feedback to **DistanceEstimator** in a given round t, we write $v^t = 1$ write $v^t = 0$ otherwise.

Definition 7. *Given an accuracy parameter ϵ, a linear parameter vector α, a sequence of vectors $u^1, \ldots, u^T$, a sequence of guesses $g^1, \ldots, g^T$ and a sequence of feedback indicators, $v^1, \ldots, v^T$, the number of valid ϵ-mistakes made by* **DistanceEstimator** *is:*

$$\textbf{Mistakes}(\epsilon) = \sum_{t=1}^{T} \mathbb{1}(v^t = 1 \wedge |g^t - \langle u^t, \alpha \rangle| > \epsilon)$$

In other words, it is the number of ϵ-mistakes made by **DistanceEstimator** *in rounds for which we provided the algorithm feedback.*

We now state a version of the main theorem from Lobel et al. [2017], adapted to our setting[4]:

Lemma 1 (Lobel et al. [2017])**.** *For any $\epsilon > 0$ and any sequence of vectors $u^1, \ldots, u^T$,* **DistanceEstimator**(ϵ) *makes a bounded number of valid ϵ-mistakes.*

$$\textbf{Mistakes}(\epsilon) = O\left(m \log\left(\frac{m \cdot B_1 \cdot B_2}{\epsilon}\right)\right)$$

3.3 The Algorithm

For each pair of arms $i, j \in [k]$, our algorithm instantiates a copy of **DistanceEstimator**(ϵ^2), which we denote by **DistanceEstimator**$_{i,j}$: we also subscript all variables relevant to **DistanceEstimator**$_{i,j}$ with i, j (e.g. $u^t_{i,j}$). The underlying linear parameter vector we want to learn $\alpha = flatten(G) \in \mathbb{R}^{d^2}$, where $flatten : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \cdot n}$ maps a $m \times n$ matrix to a vector of size mn by concatenating its rows into a vector. Similarly, given a pair of contexts x^t_i, x^t_j, we will define $u^t_{i,j} = flatten((x^t_i - x^t_j)(x^t_i - x^t_j)^\top)$. **DistanceEstimator**$_{i,j}.guess(u^t_{i,j})$ will output guess $g^t_{i,j}$ for the value $\langle \alpha, u^t_{i,j} \rangle = (\bar{d}^t_{i,j})^2$, as

$$\langle flatten(G), flatten((x^t_i - x^t_j)(x^t_i - x^t_j)^\top) \rangle = \sum_{a,b=1}^{d} G_{a,b}(x^t_i - x^t_j)_a (x^t_i - x^t_j)_b = (\bar{d}^t_{i,j})^2$$

We take $\hat{d}^t_{i,j} = \sqrt{g^t_{i,j}}$ as our estimate for the distance between x^t_i and x^t_j.

The algorithm then chooses an arm to pull according to the distribution $\pi(\bar{r}^t, \hat{d}^t)$, where $\bar{r}^t_i = \langle \theta, x_i \rangle$. The fairness oracle O_d returns all pairs of arms that violate the fairness constraints. For these pairs (i, j) we provide feedback to **DistanceEstimator**$_{i,j}$: the guess was too large. For the remaining pairs of arms (i, j), there are two cases. If the (i, j)th constraint in $LP(\bar{r}^t, \hat{d}^t)$ was not tight, then we provide no feedback ($v^t_{i,j} = 0$). Otherwise, we provide feedback: the guess was not too large. The pseudocode appears as Algorithm 1.

First we derive the valid mistake bound that the **DistanceEstimator**$_{i,j}$ algorithms incur in our parameterization.

Lemma 2. *For pair (i, j), the total number of valid ϵ^2 mistakes made by* **DistanceEstimator**$_{i,j}$ *is bounded as:*

$$\textbf{Mistakes}(\epsilon^2) = O\left(d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$$

where the distance function is defined as $d(x_i, x_j) = \|Ax_i - Ax_j\|_2$ and $\|\cdot\|_F$ denotes the Frobenius norm.

[4] In Lobel et al. [2017], the algorithm receives feedback in every round, and the scale parameters B_1 and B_2 are normalized to be 1. But the version we state is an immediate consequence.

```
for i, j = 1,...,k do
    DistanceEstimator_{i,j} = DistanceEstimator(ε^2)
end
for t = 1,...,T do
    receive the contexts x^t = (x_1^t,...,x_k^t)
    for i, j = 1,...,k do
        u_{i,j}^t = flatten((x_i^t − x_j^t)(x_i^t − x_j^t)^⊤)
        g_{i,j}^t = DistanceEstimator_{ij}.guess(u_{i,j}^t)
        d̂_{i,j}^t = √(g_{i,j}^t)
    end
    π^t = π(r̄^t, d̂^t)
    Pull an arm i^t according to π^t and receive a reward r_{i^t}^t
    S = O_d(x^t, π^t)
    R = {(i, j) | (i, j) ∉ S ∧ |p_i^t − p_j^t| = d̂_{ij}^t}
    for (i, j) ∈ S do
        DistanceEstimator_{ij}.feedback(⊥)
        v_{ij}^t = 1
    end
    for (i, j) ∈ R do
        DistanceEstimator_{ij}.feedback(⊤)
        v_{ij}^t = 1
    end
end
```

Algorithm 1: $L_{\text{known}-\theta}$

Proof. This follows directly from Lemma 1, and the observations that in our setting, $m = d^2$, $B_1 = \|\alpha\| = \|A^\top A\|_F$, and

$$B_2 \leq \max_t \|u_{i,j}^t\|_2 \leq \max_t \|(x_i^t - x_j^t)\|^2 \leq 4.$$

□

We next observe that since we only instantiate k^2 copies of **DistanceEstimator** in total, Lemma 2 immediately implies the following bound on the total number of rounds in which *any* distance estimator that receives feedback provides us with a distance estimate that differs by more than ϵ from the correct value:

Corollary 1. *The number of rounds where there exists a pair (i, j) such that feedback is provided ($v_{i,j}^t = 1$) and its estimate is off by more than ϵ is bounded:*

$$\left|\{t : \exists (i,j) : v_{ij}^t = 1 \wedge |\hat{d}_{i,j}^t - \bar{d}_{i,j}^t| > \epsilon\}\right| \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$$

Proof. This follows from summing the k^2 valid ϵ^2 mistake bounds for each copy of **DistanceEstimator**$_{i,j}$, and noting that an ϵ mistake in predicting the value of $\bar{d}_{i,j}^t$ implies an ϵ^2 mistake in predicting the value of $(\bar{d}_{i,j}^t)^2$. □

We now have the pieces to bound the ϵ-unfairness loss of our algorithm:

Theorem 1. *For any sequence of contexts and any Mahalanobis distance $d(x_1, x_2) = \|Ax_1 - Ax_2\|_2$:*

$$\textbf{FairnessLoss}(\boldsymbol{L}_{known-\theta}, T, \epsilon) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^T A\|_F}{\epsilon}\right)\right)$$

Proof.

$$\begin{aligned}
\textbf{FairnessLoss}(\boldsymbol{L}_{\text{known}-\theta}, T, \epsilon) &= \sum_{t=1}^{T} \textbf{Unfair}(\boldsymbol{L}_{\text{known}-\theta}, \epsilon) \\
&\leq \sum_{t=1}^{T} \sum_{i,j} \mathbb{1}(|\pi_i^t - \pi_j^t| > \bar{d}_{ij}^t + \epsilon) \\
&= \sum_{i,j} \sum_{t=1}^{T} \mathbb{1}(\{v_{ij}^t = 1 \wedge \hat{d}_{ij}^t > d_{ij}^t + \epsilon\}) \\
&\leq \sum_{i,j} \sum_{t=1}^{T} \mathbb{1}(\{v_{ij}^t = 1 \wedge |\hat{d}_{ij}^t - d_{ij}^t| > \epsilon\}) \\
&= O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right) && \text{Corollary 1}
\end{aligned}$$

□

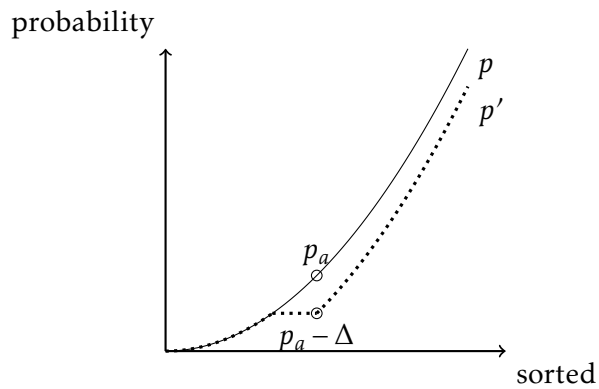

Figure 1: A visual interpretation of the surgery performed on p in the proof of Lemma 3 to obtain P'. Note that the surgery manages to shrink the distance between p_a and p_b without increasing the distance between any other pair of points.

We now turn our attention to bounding the regret of the algorithm. Recall from the overview in Section 3.1, that our plan will be to divide rounds into two types. In rounds of the first type, our distance estimates corresponding to every *tight constraint* in the linear program have only small error. We cannot bound the number of such rounds, but we can bound the regret incurred in any such rounds. In rounds of the second type, we have at least one significant error in the distance estimate corresponding to a tight constraint. We might incur significant regret in such rounds, but we can bound the number of such rounds.

The following lemma bounds the *decrease* in expected per-round reward that results from under-estimating a *single* distance constraint in our linear programming formulation.

Lemma 3. *Fix any vector of distance estimates d and any vector of rewards r. Fix a constant ϵ and any pair of coordinates $(a,b) \in [k] \times [k]$. Let d' be the vector such that $d'_{ab} = d_{ab} - \epsilon$ and $d'_{ij} = d_{ij}$ for $(i,j) \neq (a,b)$, then $\langle r, \pi(r,d)\rangle - \langle r, \pi(r,d')\rangle \leq \epsilon \sum_{i=1}^{k} r_i$*

Proof. The plan of the proof is to start with $\pi(r,d)$ and perform surgery on it to arrive at a new probability distribution $p' \in \Delta k$ that satisfies the constraints of $LP(r,d')$, and obtains objective value at least $\langle r, p'\rangle \geq \langle r, \pi(r,d)\rangle - \epsilon \sum_{i=1}^{k} r_i$. Because p' is feasible, it lower bounds the objective value of the optimal solution $\pi(r,d')$, which yields the theorem.

To reduce notational clutter, for the rest of the argument we write p to denote $\pi(r,d)$. Without loss of generality, we assume that $p_a \geq p_b$. If $p_a - p_b \leq d_{ab} - \epsilon$, then p_i is still a feasible solution to $LP(r,d')$, and we are done. Thus, for the rest of the argument, we can assume that $p_a - p_b > d_{ab} - \epsilon$. We write $\Delta = (p_a - p_b) - (d_{ab} - \epsilon) > 0$

We now define our modified distribution p':

$$p'_i = \begin{cases} p_i - \Delta & p_a \leq p_i \\ p_a - \Delta & p_a - \Delta \leq p_i < p_a \\ p_i & \text{otherwise} \end{cases}$$

We'll partition the coordinates of p_i into which of the three cases they fall into in our definition of p' above. $S_1 = \{i | p_a \leq p_i\}$, $S_2 = \{i | p_a - \epsilon \leq p_i < p_a\}$, and $S_3 = \{i | i < p_b + (d_{ab} - \epsilon)\}$. It remains to verify that p' is a feasible solution to $LP(r,d')$, and that it obtains the claimed objective value.

Feasibility: First, observe that $\sum_i p'_i \leq 1$. This follows because p' is coordinate-wise smaller than p, and by assumption, p was feasible. Thus, $\sum_i p'_i \leq \sum_i p_i \leq 1$.

Next, observe that by construction, $p'_i \geq 0$ for all i. To see this, first observe that $p_a - \Delta = p_b + (d_{ab} - \epsilon) \geq 0$ where the last inequality follows because $d_{ab} \geq \epsilon$. We then consider the three cases:

1. For $i \in S_1$, $p'_i = p_i - \Delta \geq p_a - \Delta \geq 0$ because $p_i \geq p_a$.
2. For $i \in S_2$, $p'_i = p_a - \Delta \geq 0$.
3. For $i \in S_3$, $p'_i = p_i \geq 0$.

Finally, we verify that for all (i,j), $|p'_i - p'_j| \leq d'_{ij}$. First, observe that $p'_a - p'_b = (p_b + (d_{ab} - \epsilon)) - p'_b = d_{ab} - \epsilon = d'_{ab}$, and so the inequality is satisfied for index pair (a,b). For all the other pairs $(i,j) \neq (a,b)$, we have $d'_{ij} = d_{ij}$, so it is enough to show that $|p'_i - p'_j| \leq d_{ij}$. Note that for all $x, y \in \{1,2,3\}$ with $x < y$, if $i \in S_x$ and $j \in S_y$, we have that $x \leq y$. Therefore, it is sufficient to verify the following six cases:

1. $i \in S_1, j \in S_1$: $|p'_i - p'_j| = (p_i - \Delta) - (p_j - \Delta) = p_i - p_j \leq d_{ij}$
2. $i \in S_1, j \in S_2$: $|p'_i - p'_j| = (p_i - \Delta) - (p_a - \Delta) = p_i - p_a < p_i - p_j \leq d_{ij}$
3. $i \in S_1, j \in S_3$: $|p'_i - p'_j| = (p_i - \Delta) - p_j = (p_i - p_j) - \Delta \leq (p_i - p_j) \leq d_{ij}$
4. $i \in S_2, j \in S_2$: $|p'_i - p'_j| = (p_a - \Delta) - (p_a - \Delta) = 0 \leq d_{ij}$
5. $i \in S_2, j \in S_3$: $|p'_i - p'_j| = (p_a - \Delta) - p_j \leq p_i - p_j \leq d_{ij}$
6. $i \in S_3, j \in S_3$: $|p'_i - p'_j| = p_i - p_j \leq d_{ij}$

Thus, we have shown that p' is a feasible solution to $LP(r, d')$.

Objective Value: Note that for each index i, $p_i - p'_i \leq \Delta \leq \epsilon$. Therefore we have:

$$\begin{aligned}
\langle r, \pi(r,d) \rangle - \langle r, \pi(r,d') \rangle &\leq \langle r, \pi(r,d) \rangle - \langle r, p' \rangle \\
&= \langle r, p - p' \rangle \\
&\leq \epsilon \sum_{i=1}^{k} r_i
\end{aligned}$$

which completes the proof. □

We now prove the main technical lemma of this section. It states that in any round in which the error of our distance estimates for *tight constraints* is small (even if we have high error in the distance estimates for slack constraints), then we will have low per-round regret.

Lemma 4. *At round t, if for all pairs of indices (i,j), we have either:*

1. $|\hat{d}^t_{i,j} - \bar{d}^t_{i,j}| \leq \epsilon$ *or*

2. $v^t_{i,j} = 0$ *(corresponding to an LP constraint that is not tight)*

then:

$$\langle r^t, \pi(r^t, \bar{d}^t)\rangle - \langle r^t, \pi(r^t, \hat{d}^t)\rangle \leq \epsilon k^3$$

for any vector r^t *with* $\|r^t\|_\infty \leq 1$.

Proof. First, define $\tilde{d}^t$ to be the coordinate-wise maximum of $\hat{d}^t$ and $\bar{d}^t$: i.e. the vector such that for every pair of coordinates i, j, $\tilde{d}_{ij} = \max(\bar{d}_{ij}, \hat{d}_{ij})$. To simplify notation, we will write $\hat{p} = \pi(r^t, \hat{d}^t)$, $\bar{p} = \pi(r^t, \bar{d}^t)$, and $\tilde{p} = \pi(r^t, \tilde{d}^t)$.

We make three relevant observations:

1. First, because $LP(r^t, \tilde{d}^t)$ is a relaxation of $LP(r^t, \bar{d}^t)$, it has only larger objective value. In other words, we have that $\langle r^t, \tilde{p}\rangle \geq \langle r^t, \bar{p}\rangle$. Thus, it suffices to prove that $\langle r^t, \hat{p}\rangle \geq \langle r^t, \tilde{p}\rangle - \epsilon k^3$.

2. Second, for all pairs i, j, $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq |\hat{d}^t_{i,j} - \bar{d}^t_{i,j}|$. Thus, if we had $|\hat{d}^t_{i,j} - \bar{d}^t_{i,j}| \leq \epsilon$, we also have $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq \epsilon$.

3. Finally, by construction, for every pair (i, j), we have $\tilde{d}_{ij} \geq \hat{d}_{ij}$

Let S_1 be the set of indices (i, j) such that $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq \epsilon$, and let S_2 be the set of indices $(i, j) \notin S_1$ such that $v^t_{i,j} = 0$. Note that by assumption, these partition the space, and that by construction, for every $(i, j) \in S_2$, the corresponding constraint in $LP(r^t, \hat{d}^t)$ is not tight: i.e. $|\hat{p}_i - \hat{p}_j| < \hat{d}^t_{i,j}$. Let d^* be the vector such that for all $(i, j) \in S_1$, $d^*_{ij} = \hat{d}_{ij}$, and for all $(i, j) \in S_2$, $d^*_{ij} = \tilde{d}_{ij}$. Observe that $LP(r^t, d^*)$ corresponds to a relaxation of $LP(r^t, \hat{d})$ in which *only constraints that were already slack were relaxed*. As a result, $\hat{p}$ is also an optimal solution to $LP(r^t, d^*)$. Note also that by construction, we now have that for *every* pair (i, j): $|\tilde{d}_{ij} - d^*_{ij}| \leq \epsilon$

Our argument will proceed by describing a sequence of $n + 1 = k^2 + 1$ vectors $p^0, p^1, \ldots, p^n$ such that $p^0 = \tilde{p}$, p^n is a feasible solution to $LP(r^t, d^*)$, and for all adjacent pairs $p^\ell, p^{\ell+1}$, we have: $\langle r^t, p^{\ell+1}\rangle \geq \langle r^t, p^\ell\rangle - \epsilon k$. Telescoping these inequalities yields:

$$\langle r^t, \hat{p}\rangle \geq \langle r^t, p^n\rangle \geq \langle r^t, \tilde{p}\rangle - k^3\epsilon$$

which will complete the proof.

To finish the argument, fix an arbitrary ordering on the indices $(i, j) \in [k]\times[k]$, which we denote by $(i_1, j_1), \ldots, (i_n, j_n)$. Define the distance vector d^ℓ such that:

$$d^\ell_{i_a, j_a} = \begin{cases} \tilde{d}_{i_a, j_a}, & \text{If } a > \ell; \\ d^*_{i_a, j_a}, & \text{If } a \leq \ell. \end{cases}$$

Note that the sequence of distance vectors $d^1, \ldots, d^n$ "walks between" $\tilde{d}$ and d^* one coordinate at a time. Now let $p^\ell = \pi(r^t, d^\ell)$. By construction, we have that every pair $(d^\ell, d^{\ell+1})$ differ in only a single coordinate, and that the difference has magnitude at most ϵ. Therefore, we can apply Lemma 3 to conclude that:

$$\langle r^t, p^{\ell+1}\rangle \geq \langle r^t, p^\ell\rangle - \epsilon \sum_{i=1}^{k} r^t_i \geq \langle r^t, p^\ell\rangle - \epsilon k$$

as desired. □

Finally, we have all the pieces we need to prove a regret bound for $\boldsymbol{L}_{\text{known}-\theta}$.

Theorem 2. *For any time horizon T:*

$$\textbf{Regret}(\boldsymbol{L}_{known-\theta}, T) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right) + k^3 \epsilon T\right)$$

Setting $\epsilon = O(1/(k^3 T))$ yields a regret bound of $O(d^2 \log(\|A^\top A\|_F \cdot dkT))$.

Proof. We partition the rounds t into two types. Let S_1 denote the rounds such that there is at least one pair of indices (i, j) such that one instance **DistanceEstimator**$_{ij}$ produced an estimate that had error more than ϵ, and it was provided feedback. We let S_2 denote the remaining rounds, for which for *every* pair of indices (i, j), *either* **DistanceEstimator**$_{ij}$ produced an estimate that had error at most ϵ, or **DistanceEstimator**$_{ij}$ was not given feedback.

$$S_1 = \{t : \exists (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| > \epsilon \text{ and } v_{ij}^t = 1\} \quad S_2 = \{t : \forall (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| \leq \epsilon \text{ or } v_{ij}^t = 0\}$$

Observe that S_1 and S_2 partition the set of all rounds. Next, observe that Corollary 1 tells us that:

$$|S_1| \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$$

and Lemma 4 tells us that for every round $t \in S_2$, the per-round regret is at most ϵk^3. Together with the facts that $|S_2| \leq T$ and that the per-round regret for any $t \in S_1$ is at most 1, we obtain:

$$\textbf{Regret}(\boldsymbol{L}_{\text{known}-\theta}, T) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right) + k^3 \epsilon T\right)$$

□

4 The Full Algorithm

In this section, we present our final algorithm, which has no knowledge of either the distance function d or the linear objective θ. The resulting algorithm shares many similarities with the algorithm we developed in Section 3, and so much of the analysis can be reused.

4.1 Outline of the Solution

At a high level, our plan will be to combine the techniques we developed in Section 3 with a standard "optimism in the face of uncertainty" strategy for learning the parameter vector θ. Our algorithm will maintain a ridge-regression estimate $\tilde{\theta}$ together with confidence regions derived in Abbasi-Yadkori et al. [2011]. After it observes the contexts x_i^t at round t, it uses these to derive upper confidence bounds on the expected rewards, corresponding to each context — represented as a vector $\hat{r}^t$. The algorithm continues to maintain distance estimates $\hat{d}^t$ using the **DistanceEstimator** subroutines, identically to how they were used in Section 3. At ever round, the algorithm then chooses its action according to the distribution $\pi^t = \pi(\hat{r}^t, \hat{d}^t)$.

The regret analysis of the algorithm follows by decomposing the per-round regret into two pieces. The first can be bounded by the sum of the *expected widths* of the confidence intervals

corresponding to each context x_i^t that might be chosen at each round t, where the expectation is over the randomness of the algorithm's distribution π^t. A theorem of Abbasi-Yadkori et al. [2011] bounds the sum of the widths of the confidence intervals corresponding to arms *actually chosen* by the algorithm (Lemma 6). Using a martingale concentration inequality, we are able to relate these two quantities (Lemma 8). We show that the second piece of the regret bound can be manipulated into a form that can be bounded using Lemmas 1 and 4 from Section 3 (Theorem 3).

4.2 Confidence Intervals from Abbasi-Yadkori et al. [2011]

We would like to be able to construct confidence intervals at each round t around each arm's expected reward such that for each arm i, with probability $1-\delta$, $\bar{r}_i^t \in [\tilde{r}_i^t + w_i^t, \tilde{r}_i^t + w_i^t]$, where $\tilde{r}_i^t$ is our ridge-regression estimate of $\bar{r}_i^t$ and w_i^t is the confidence interval width around the estimate. Our algorithm will make use of such confidence intervals for the ridge regression estimator derived and analyzed in Abbasi-Yadkori et al. [2011], which we recount here.

Let $\tilde{V}^t = X^{t\top} X^t + \lambda I$ be a regularized design matrix, where $X^t = [x_{i_1}^1, \ldots, x_{i_{t-1}}^{t-1}]$ represents all the contexts whose rewards we have observed up to but not including time t. Let $Y^t = [r_{i_1}^1, \ldots, r_{i_{t-1}}^{t-1}]$ be the corresponding vector of observed rewards. $\tilde{\theta} = (V^t)^{-1} X^{t\top} Y^t$ is the (ridge regression) regularized least squares estimator we use at time t. We write $\tilde{r}_i^t = \langle \tilde{\theta}, x_i^t \rangle$ for the reward point prediction that this estimator makes at time t for arm i.

We can construct the following confidence intervals around $\tilde{r}^t$:

Lemma 5 (Abbasi-Yadkori et al. [2011]). *With probability* $1-\delta$,

$$|\bar{r}_i^t - \tilde{r}_i^t| = |\langle x_i^t, (\theta - \tilde{\theta}) \rangle| \leq \|x_i^t\|_{(\bar{V}^t)^{-1}} \left(\sqrt{2d \log(\frac{1 + t/\lambda}{\delta})} + \sqrt{\lambda} \right)$$

where $\|x\|_A = \sqrt{x^\top A x}$

Therefore, the confidence interval widths we use in our algorithm will be

$$w_i^t = \min(\|x_i^t\|_{(\bar{V}^t)^{-1}} \left(\sqrt{2d \log(\frac{1 + t/\lambda}{\delta})} + \sqrt{\lambda} \right), 1)$$

(expected rewards are bounded by 1 in our setting, and so the minimum maintains the validity of the confidence intervals). The upper confidence bounds we use to compute our distribution over arms will be $\hat{r}_i^t = \tilde{r}_i^t + w_i^t$. We will write $w^t = [w_1^t, \ldots, w_k^t]$ to denote the vector of confidence interval widths at round t.

Little can be said about the widths of these confidence intervals in isolation. However, the following theorem bounds the *sum* (over time) of the widths of the confidence intervals around the contexts actually selected.

Lemma 6 (Abbasi-Yadkori et al. [2011]).

$$\sum_{t=1}^{T} w_{i^t}^t \leq \sqrt{2d \log\left(1 + \frac{T}{d\lambda}\right)} \left(\sqrt{2dT \log(\frac{1 + T/\lambda}{\delta})} + \sqrt{T\lambda} \right)$$

for $i, j = 1, \ldots, k$ **do**
  DistanceEstimator$_{ij}$ = **DistanceEstimator**(ϵ^2)
end
for $t = 1, \ldots, T$ **do**
  receive the contexts $x^t = (x_1^t, \ldots, x_k^t)$
  $X^t = [x^1, \ldots, x^{t-1}]$
  $Y^t = [r^t, \ldots, r^{t-1}]$
  $\tilde{V}^t = X^{t\top} X^t + \lambda I$
  $\tilde{\theta} = (V^t)^{-1} X^{t\top} Y^t$
  for $i = 1, \ldots, k$ **do**
    $\tilde{r}_i^t = \langle \tilde{\theta}, x_i^t \rangle$
    $w_i^t = \min\left(\|x_i^t\|_{(\bar{V}^t)^{-1}} \left(\sqrt{2d \log(\frac{1+t/\lambda}{\delta})} + \sqrt{\lambda} \right), 1 \right)$
    $\hat{r}_i^t = \tilde{r}_i^t + w_i^t$
  end
  for $i, j = 1, \ldots, k$ **do**
    $u_{i,j}^t = flatten((x_i^t - x_j^t)(x_i^t - x_j^t)^T))$
    $g_{i,j}^t$ = **DistanceEstimator**$_{i,j}.guess(u_{i,j}^t)$
    $\hat{d}_{ij}^t = \sqrt{g_{i,j}^t}$
  end
  $\pi^t = \pi(\hat{r}^t, \hat{d}^t)$
  Pull an arm i^t according to π^t and receive a reward $r_{i^t}^t$
  $S = \boldsymbol{O}_d(x^t, \pi^t)$
  $R = \{(i,j) | (i,j) \notin S \wedge |\pi_i^t - \pi_j^t| = \hat{d}_{i,j}^t\}$
  for $(i, j) \in S$ **do**
    DistanceEstimator$_{i,j}.feedback(\perp)$
    $v_{i,j}^t = 1$
  end
  for $(i, j) \in R$ **do**
    DistanceEstimator$_{i,j}.feedback(\top)$
    $v_{i,j}^t = 1$
  end
end

Algorithm 2: $\boldsymbol{L}_{\text{full}}$

4.3 The Algorithm

The pseudocode for the full algorithm is given in Algorithm 2.

In our proof of Theorem 3, we will connect the regret of $\boldsymbol{L}_{full}$ to the sum of the *expected* widths of the confidence intervals pulled at each round. In contrast, what is bounded by Lemma 6 is the sum of the *realized* widths. Using the Azuma Hoeffding inequality, we can relate these two quantities.

Lemma 7 (Azuma-Hoeffding inequality (Hoeffding [1963])). *Suppose* $\{X_k : k = 0,1,2,3,\ldots\}$ *is a martingale and*

$$\left|X_k - X_{k-1}\right| < c_k.$$

Then, for all positive integers N *and all positive reals* t*,*

$$\Pr(X_N - X_0 \geq t) \leq \exp\left(\frac{t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

Lemma 8.

$$\Pr\left(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \sqrt{2T\log\frac{1}{\delta}}\right) \leq \delta$$

Proof. Once $x^1,\ldots,x^{t-1}, r_{i^t}^1,\ldots,r_{i^{t-1}}^{t-1}$ and x^t are fixed, π^t is fixed. In other words, for the filtration $\mathscr{F}^t = \sigma(x^1,\ldots,x^{t-1}, r_{i^t}^1,\ldots,r_{i^{t-1}}^{t-1}, x^t)$, $w_{i^t}^t$ is $\mathscr{F}^t$ measurable. Now, define

$$D^t = \sum_{s=1}^{t} \mathbb{E}_{i\sim\pi^s}[w_i^s] - \sum_{s=1}^{t} w_{i^s}^s$$

with respect to $\mathscr{F}^t$. One can think of D^t as the accumulated difference between the confidence width of the arm that was actually pulled and the expected confidence width. It's easy to see that $\{D^t\}$ is a martingale, as $\mathbb{E}[D^1] = 0$, and $\mathbb{E}[D^{t+1}|\mathscr{F}^t] = D^t$.

Also, $D_t - D_{t-1} = w_{i^t}^t - \mathbb{E}_{i\sim\pi^t}[w_i^t] \leq 1$, since the confidence interval widths are bounded by 1.

Applying the Azuma-Hoeffding inequality gives us the following:

$$\Pr\left(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \epsilon\right) = \Pr(D^T \geq \epsilon) \leq \exp\left(\frac{-\epsilon^2}{2T}\right)$$

Now, setting $\epsilon = \sqrt{2T\ln\frac{1}{\delta}}$ yields:

$$\Pr\left(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \sqrt{2T\log\frac{1}{\delta}}\right) \leq \delta$$

□

Theorem 3. *For any time horizon* T*, with probability* $1-\delta$*:*

$$\textbf{Regret}(\boldsymbol{L}_{full}, T) \leq O\left(k^2 d^2 \log\left(\frac{d\cdot\|A^\top A\|_F}{\epsilon}\right) + k^3\epsilon T + d\sqrt{T}\log\left(\frac{T}{\delta}\right)\right)$$

If $\epsilon = 1/k^3 T$*, this is a regret bound of* $O\left(k^2 d^2 \log\left(kdT\cdot\|A^\top A\|_F\right) + d\sqrt{T}\log\left(\frac{T}{\delta}\right)\right)$

Proof. We can compute:

$$\begin{aligned}
\textbf{Regret}(\boldsymbol{L}_{full}, T) &= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\pi(\bar{r}^t,\bar{d}^t)}[\bar{r}_i^t] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\pi(\hat{r}^t,\hat{d}^t)}[\bar{r}_i^t] \\
&= \sum_{t=1}^{T} \langle \bar{r}^t, \pi(\bar{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&= \sum_{t=1}^{T} \langle \bar{r}^t, \pi(\bar{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq \sum_{t=1}^{T} \langle \hat{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq \sum_{t=1}^{T} \langle 2w^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle
\end{aligned}$$

Here, the first inequality follows from the fact that $\hat{r}^t$ is coordinate-wise larger than $\bar{r}^t$, and that $\pi(\hat{r}^t, \bar{d}^t)$ is the optimal solution to $LP(\hat{r}^t, \bar{d}^t)$. The second inequality follows from $\bar{r} \in [\tilde{r} - w, \tilde{r} + w] = [\hat{r} - 2w, \hat{r}]$.

Just as in the proof of Theorem 2, we now partition time into two sets:

$$S_1 = \{t : \exists (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| > \epsilon \text{ and } v_{ij}^t = 1\} \quad S_2 = \{t : \forall (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| \leq \epsilon \text{ or } v_{ij}^t = 0\}$$

Recall that corollary 1 bounds $|S_1| \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$. Since the per-step regret of our algorithm can be at most 1, this means that rounds $t \in S_1$ can contribute in total at most $C \doteq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$ regret. Thus, for the rest of our analysis, we can focus on rounds $t \in S_2$.

Fix any round $t \in S_2$. From Lemma 4 we have:.

$$\langle \hat{r}, \pi(\hat{r}, \bar{d})\rangle - \langle \hat{r}, \pi(\hat{r}, \hat{d})\rangle \leq k^3 \epsilon$$

Further manipulations give:

$$\begin{aligned}
\left(\langle \hat{r}, \pi(\hat{r}, \bar{d})\rangle - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle\right) - \left(\langle \hat{r}, \pi(\hat{r}, \hat{d})\rangle - \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle\right) &\leq k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle \\
\langle 2w, \pi(\hat{r}, \bar{d})\rangle - \langle 2w, \pi(\hat{r}, \hat{d})\rangle &\leq k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, d)\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle \\
\langle 2w, \pi(\hat{r}, \bar{d})\rangle &\leq \langle 2w, \pi(\hat{r}, \hat{d})\rangle + k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle
\end{aligned}$$

Now, substituting the above expressions back into our expression for regret:

$$
\begin{aligned}
&\textbf{Regret}(\boldsymbol{L}_{full}, T) \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \bar{d}^t) \rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t) \rangle - \langle \bar{r}_i^t, \pi(\hat{r}^t, \hat{d}^t) \rangle \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \hat{d}^t) \rangle + k^3\epsilon - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t) \rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t) \rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t) \rangle - \langle \bar{r}_i^t, \pi(\hat{r}^t, \hat{d}^t) \rangle \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \hat{d}^t) \rangle + k^3\epsilon \\
&\leq C + 2\sum_{t \in S_2} \mathop{\mathbb{E}}_{i \in \pi(\hat{r}^t, \hat{d}^t)} [w_i^t] + k^3\epsilon \\
&\leq C + k^3\epsilon T + 2\left(\sqrt{2d\log\left(1 + \frac{T}{d\lambda}\right)}\left(\sqrt{2dT\log(\frac{1+T/\lambda}{\delta})} + \sqrt{T\lambda}\right) + \sqrt{2T\log\frac{1}{\delta}}\right) \\
&= O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right) + k^3\epsilon T + O(d\sqrt{T}\log(\frac{T}{\delta}))
\end{aligned}
$$

The last inequality holds with probability $1-\delta$ and uses Lemmas 6 and 8, and sets $\lambda = 1$.

□

Finally, the bound on the fairness loss is identical to the bound we proved in Theorem 1 (because our algorithm for constructing distance estimates $\hat{d}$ is unchanged). We have:

Theorem 4. *For any sequence of contexts and any Mahalanobis distance $d(x_1, x_2) = \|Ax_1 - Ax_2\|_2$:*

$$\textbf{FairnessLoss}(\boldsymbol{L}_{full}, T, \epsilon) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$$

5 Conclusion and Future Directions

We have initiated the study of fair sequential decision making in settings where the notions of payoff and fairness are separate and may be in tension with each other, and have shown that in a stylized setting, optimal fair decisions can be efficiently learned *even without direct knowledge of the fairness metric*. A number of extensions of our framework and results would be interesting to examine. At a high level, the interesting question is: how much can we further relax the information about the fairness metric available to the algorithm? For instance, what if the fairness feedback is only partial, identifying some but not all fairness violations? What if it only indicates whether or not there were any violations, but does not identify them? What if the feedback is not guaranteed to be exactly consistent with any metric? Or what if the feedback is consistent with *some* distance function, but not one in a known class: for example, what if the distance is not exactly Mahalanobis, but is approximately so? In general, it is very interesting to continue to push to close the wide gap between the study of individual fairness notions and the study of group fairness notions. When can we obtain the strong semantics of individual fairness without making correspondingly strong assumptions?

Acknowledgements

We thank Steven Wu and Matthew Joseph for helpful discussions at an early stage of this work.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2312–2320, 2011. URL http://papers.nips.cc/paper/4417-improved-algorithms-for-linear-stochastic-bandits.

Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: the state of the art. *arXiv preprint arXiv:1703.09207*, 2017.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv preprint arXiv:1703.00056*, 2017.

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.

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

Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 25(7):1445–1459, 2013.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 2016.

Ursula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Calibration for the (computationally-identifiable) masses. *arXiv preprint arXiv:1711.08513*, 2017.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *International Conference on Machine Learning*, pages 1617–1626, 2017.

Prateek Jain, Brian Kulis, Inderjit S Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in neural information processing systems*, pages 761–768, 2009.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. pages 325–333, 2016a.

Matthew Joseph, Michael J. Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Fair algorithms for infinite and contextual bandits. *CoRR*, abs/1610.09559, 2016b. URL http://arxiv.org/abs/1610.09559.

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

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv preprint arXiv:1711.05144*, 2017.

Michael P Kim, Omer Reingold, and Guy N Rothblum. Fairness through computationally-bounded awareness. *arXiv preprint arXiv:1803.03239*, 2018.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science, Berkeley, CA, USA, 2017*, 2017.

Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4): 287–364, 2013.

Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C Parkes. Calibrated fairness in bandits. *arXiv preprint arXiv:1707.01875*, 2017.

Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, page 585, 2017. doi: 10.1145/3033274.3085100. URL http://doi.acm.org/10.1145/3033274.3085100.

Guy N Rothblum and Gal Yona. Probably approximately metric-fair learning. *arXiv preprint arXiv:1803.03242*, 2018.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017.

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

A Generalization to Multiple Actions

In the body of the paper, we analyzed the standard contextual bandit setting in which the algorithm must choose a *single* action to take at each round. However, it is often the case that this constraint is artificial and undesirable in settings for which fairness is a concern. Consider, for example, the case of lending: at each round, a bank observes the loan applications of a collection of individuals, and decides whom to grant loans to. Some loans may be profitable and some loans may not be — so the optimal policy is non-trivial. But there need not be a budget constraint — the optimal policy may grant loans to as many qualified individuals as there are on a given day.

In our framework, this corresponds to letting the algorithm take as many as k actions on a single day. Fortunately, all of our results generalize to this case. The maximum reward per day in this case increases from 1 to k, so naturally the regret bound we obtain is also a factor of k larger. In this section, we explain the details of our proof that need to be modified.

The first step is to consider a modified linear program $LP(a,c)$, which we will write as $LP_m(a,c)$. It simply replaces the simplex constraint that the probabilities of actions sum to 1 with the hypercube constraint that no probability can be greater than 1:

$$\begin{aligned} \underset{\pi=\{p_1,\ldots,p_k\}}{\text{maximize}} \quad & \sum_{i=1}^{k} p_i a_i \\ \text{subject to} \quad & |p_i - p_j| \leq c_{i,j}, \forall (i,j) \\ & 0 \leq p_i \leq 1, \forall i \end{aligned}$$

We must also change our definition of regret, because the benchmark we want to compete with is the best fair policy that can make up to k action selections per round. This simply corresponds to comparing to a benchmark which is defined with respect to $LP_m(a,c)$ — but the form of the regret is unchanged:

$$\begin{aligned} & \textbf{Regret}_m(\boldsymbol{L}, T) \\ &= \sum_{t=1}^{T} \sum_{i=1}^{k} \bar{r}_i^t \cdot Pr(\text{best fair policy pulls arm } i \text{ in round } t) - \bar{r}_i^t \cdot Pr(\boldsymbol{L} \text{ pulls arm } i \text{ in round } t) \\ &= \sum_{t=1}^{T} \langle \bar{r}^t, \pi(\bar{r}^t, \bar{d}^t) \rangle - \langle \bar{r}^t, f^t(h^t, x^t) \rangle \end{aligned}$$

where π is defined exactly as before, except with respect to $LP_m(a,c)$.

The first observation is that our generalization to multiple arms does not affect our analysis of fairness loss at all, since we are able to bound this without reference to the rewards. That is, we still have that fairness loss is bounded as

$$\textbf{FairnessLoss}(\boldsymbol{L}_{full_m}, T, \epsilon) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$$

As for our regret analysis, certain terms in the regret scale by a factor of k.

$$\textbf{Regret}_m(\boldsymbol{L}_{full_m}, T) \leq O\left(k^3 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right) + k^3 \epsilon T + dk\sqrt{k^2 T} \log(\frac{kT}{\delta})\right)$$

Proof. There are only two parts of our proof that depend on the structure on the linear program $LP(a,c)$. The first is the proof of Lemma 3, which uses the fact that if we take a feasible solution to $LP(a,c)$ and reduce its values pointwise, we maintain feasibility — that is, that the feasible region of $LP(a,c)$ is downward closed. But note that the feasible region of $LP_m(a,c)$ is also downward closed, so the same argument goes through. Recall that our analysis in the known objective case partitions rounds into two sorts: rounds for which we can bound our per-round regret (from

Lemma 3), and a bounded number of rounds in which we cannot. For those rounds in which we cannot bound the per-round regret, the maximum regret is now k rather than 1. So, our regret during these rounds increases by a factor of k to $O\left(k^3 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right)$.

Therefore, we have that

$$\textbf{Regret}_m(\boldsymbol{L}_{full_m}, T) \leq O\left(k^3 d^2 \log\left(\frac{d \cdot \|A^\top A\|_F}{\epsilon}\right)\right) + k^3 \epsilon T + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \hat{d}^t)\rangle$$

where $S_2 = \{t : \forall (i,j) : |\hat{d}^t_{ij} - \bar{d}^t_{ij}| \leq \epsilon \text{ or } v^t_{ij} = 0\}$

Next, we need to consider the final term in this expression. $\langle w^t, \pi(\hat{r}^t, \hat{d}^t)\rangle$ is the expected sum of the confidence interval widths of the arms that are pulled at round t. By the same martingale argument as in lemma 8, with high probability, the expected sum of the confidence interval widths over time horizon T is close to the realized sum of the confidence widths of the arms pulled; in this case, the martingale is

$$D^t = \sum_{s=1}^{t} \sum_{i=1}^{k} w_i^s \cdot \Pr(\text{arm } i \text{ is pulled in round } s) - \sum_{s=1}^{t} \sum_{i=1}^{k} w_i^s \cdot \mathbb{1}(\text{arm } i \text{ is pulled in round } s)$$

However, in this case, the martingale difference is bounded by at most k instead of 1. Hence, applying the Azuma-Hoeffding inequality gives us that with probability $1 - \delta$,

$$\sum_{t=1}^{T} \sum_{i=1}^{k} w_i^t \cdot \Pr(\text{arm } i \text{ is pulled in round } t) \leq \sum_{t=1}^{T} \sum_{i=1}^{k} w_i^t \cdot \mathbb{1}(\text{arm } i \text{ is pulled in round } t) + \sqrt{2k^2 T \log \frac{1}{\delta}}$$

First, note that the confidence interval derived from lemma 5 remains valid. Also, $\bar{V}^t = \bar{V}^{t-1} + \sum_{i \in P^t} x_i^t x_i^{t\top}$. For simplicity in notation, we write $P^t = \{i : \text{arm } i \text{ is pulled in round } t\}$. So we need to bound $\sum_{t=1}^{T} \sum_{i \in P^t} w_i^t$.

We can then derive:

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{i \in P^t} w_i^t &\leq \sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(\bar{V}^{t-1})^{-1}} \left(\sqrt{2d \log(\frac{1 + t/\lambda}{\delta})} + \sqrt{\lambda}\right) \\
&\leq \sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(\bar{V}^{t-1})^{-1}} \left(\sqrt{2d \log(\frac{1 + t/\lambda}{\delta})}\right) + \sum_{t=1}^{T} \sum_{i \in P^t} \left(\|x_i^t\|_{(\bar{V}^{t-1})^{-1}} \sqrt{\lambda}\right) \\
&\leq \sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(\bar{V}^{t-1})^{-1}} \cdot \left(\sqrt{\sum_{t=1}^{T} \sum_{i \in P^t} 2d \log(\frac{1 + t/\lambda}{\delta})}\right) + \sqrt{\sum_{t=1}^{T} \sum_{i \in P^t} \lambda} \\
&\leq \sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(\bar{V}^{t-1})^{-1}} \cdot \left(\sqrt{2dkT \log(\frac{1 + kT/\lambda}{\delta})}\right) + \sqrt{kT\lambda}
\end{aligned}$$

For each $i \in [k]$, write A_i to denote the set of rounds that arm i is pulled. $\sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(\bar{V}^t)^{-1}} = \sum_{i=1}^{k} \sum_{t \in A_i} \|x_i^t\|_{(\bar{V}^t)^{-1}}$, so for each $i \in [k]$, we'll bound $\sum_{t \in A_i} \|x_i^t\|_{(\bar{V}^t)^{-1}}$.

Lemma 9.

$$\sum_{t \in A_i} \|x_i^t\|_{(V^{\bar{t}-1})^{-1}} \leq \sqrt{2d \log\Big(1 + \frac{kT}{d\lambda}\Big)}$$

Proof. We'll iterate each $\|x_i^t\|_{(V^{\bar{t}-1})^{-1}}$ first over round $t = 1, \ldots, T$ and then $j \in P^t$ where the order of P^t has its very first element as $\|x_i^t\|$ and the rest is arbitrary. Let's call this indexing a. First, we have that $\bar{V}(a) = \bar{V}(a-1) + x(a)x(a)^\top$. More importantly, because of the way we chose to index, for each $t \in A_i$ and index a that corresponds to (i,t), $\|x_i^t\|_{(\bar{V}^{t-1})^{-1}} = \|x(a)\|_{(\bar{V}(a-1))^{-1}}$

From Lemma 11 in Abbasi-Yadkori et al. [2011] we have $\sum_{a=1}^{N} \|x(a)\|_{(\bar{V}(a-1))^{-1}} \leq \sqrt{2d \log\Big(1 + \frac{N}{d\lambda}\Big)}$, where $N \leq kT$.

Therefore, we have that

$$\sum_{t \in A_i} \|x_i^t\|_{(V^{\bar{t}-1})^{-1}} \leq \sum_{a=1}^{N} \|x(a)\|_{(\bar{V}(a-1))^{-1}} \leq \sqrt{2d \log\Big(1 + \frac{kT}{d\lambda}\Big)}$$

□

Applying the above lemma for each arm $i \in [k]$, we have

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{i \in P^t} w_i^t &\leq \sum_{t=1}^{T} \sum_{i \in P^t} \|x_i^t\|_{(V^{\bar{t}-1})^{-1}} \cdot \Big(\sqrt{2dkT \log(\frac{1 + kT/\lambda}{\delta})}\Big) + \sqrt{kT\lambda} \\
&\leq k\sqrt{2d \log\Big(1 + \frac{kT}{d\lambda}\Big)} \cdot \Big(\sqrt{2dkT \log(\frac{1 + kT/\lambda}{\delta})}\Big) + \sqrt{kT\lambda}
\end{aligned}$$

□