Manipulation-resistant Online Learning

by

Paul Christiano

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Umesh Vazirani, Chair
Peter Bartlett
Anil Aswani

Spring 2017

Manipulation-resistant Online Learning

Abstract

Manipulation-resistant Online Learning

by

Paul Christiano

Doctor of Philosophy in Computer Science

University of California, Berkeley

Umesh Vazirani, Chair

Learning algorithms are now routinely applied to data aggregated from millions of untrusted users, including reviews and feedback that are used to define learning systems' objectives. If some of these users behave manipulatively, traditional learning algorithms offer almost no performance guarantee to the "honest" users of the system. This dissertation begins to fill in this gap.

Our starting point is the traditional online learning model. In this setting a learner makes a series of decisions, receives a *loss* after each decision, and aims to achieve a total loss which is nearly as low as if they had chosen the best fixed decision-making strategy in hindsight.

We extend this model by introducing a set of users $\mathcal{U}$. Each of the learner's decisions is made on behalf of a particular user $u \in \mathcal{U}$, and u reports the loss they incur from the decision. We assume that there is some (unknown) set of "honest" users $H \subset \mathcal{U}$, who report their losses honestly, while the other users may behave adversarially. Our goal is to ensure that the total loss incurred by *users in* H is nearly as small as if all users in H had used the single best fixed decision-making strategy in hindsight. We say that an algorithm is *manipulation-resistant* if it achieves a bound of this form.

This dissertation proposes and analyzes manipulation-resistant algorithms for prediction with expert advice, contextual bandits, and collaborative filtering. These algorithms guarantee that the honest users perform nearly as well as if they had known each others' identities in advance, pooled all of their data, and then used a traditional learning algorithm. This bounds the total amount of damage that can be done per manipulative user. More significantly, we give bounds that can be considerably smaller in the realistic setting where the users are vertices of a graph (such as a social graph) with disproportionately few edges between honest and manipulative users.

As a key technical ingredient, we introduce the problem of *online local learning*, and propose a novel semidefinite programming algorithm for this problem. This algorithm allows us to effectively perform online learning over the exponentially large space of all possible sets $H \subset \mathcal{U}$, and as a side-effect provides the first asymptotically optimal algorithm for online max cut.

Contents

Chapter 1

Introduction

Today, machine learning helps us decide which products to buy, what news to read, where to stay, how to travel, and more. In modern applications, learning is often based on data aggregated from many (potentially dishonest) users. Manipulative behavior then poses a fundamental challenge to the traditional formalism of statistical learning theory: if our algorithm maximizes an objective which can itself be manipulated by dishonest users, then the guarantees of learning theory often cease to be meaningful.

For example, suppose that a marketplace uses online learning to choose which merchants to recommend, and tries to recommend merchants who will be reviewed well. A manipulative merchant could pay dishonest users for positive reviews. The learning algorithm may eventually choose to recommend the manipulative merchant: this successfully optimizes the learning algorithm's objective of recommending well-reviewed merchants, but it harms the honest users of the system and is not what the system designer intended.

Online services currently identify malicious users by behavioral cues, like the words used in a review or the age of an account. But nothing prevents manipulative users from building convincing online identities or inserting manipulative data in a way that looks authentic. The current approach leads directly to arms race between system designers and manipulative users, with no reason to expect the system designers to win in the long run. Important algorithms often literally cultivate "security by obscurity"—for example, Google executives have insisted on keeping ranking algorithms secret not because they might be used by a competitor but because they might be exploited by manipulators [21]. We propose algorithms which continue to achieve meaningful guarantees in the worst case, when they are attacked by a large number of perfectly informed and coordinated manipulators.

We consider manipulation-resistant variants of three problems in online learning: prediction with expert advice, contextual bandits, and collaborative filtering. In the manipulation-resistant version of these problems, there is a single *active* user for each decision. The learner is told the identity of the active user before making their choice. After making their decision, the learner observes a loss reported by the user. In the example of an online marketplace, a decision occurs when a user considers purchasing from a merchant. The learner is told the identity of the user, and must decide whether to recommend the purchase or recommend that the user search for an alternative. After the interaction occurs, the user reports the quality of the interaction.

In the traditional online learning setting the learner's goal is to minimize its total loss, and in particular to achieve loss nearly as low as if it had picked the best fixed strategy in hindsight.

We instead assume that there is some (unobserved) set of "honest" users, and that the goal of the learning algorithm is to minimize the total loss *in rounds involving an honest user*. More precisely, we want the total loss of the honest users to be nearly as low as if they had chosen the best fixed strategy in hindsight. This formalism for collaborative learning is novel, although it bears similarities to many strands of prior work that we discuss in Section 1.3.

Because the learning algorithm doesn't know which users are honest, it will have to incur some additional regret above and beyond what would be needed to solve the underlying learning problem. The goal is to minimize this additional regret, thereby bounding the additional damage that can be done by manipulative users.

We prove bounds on the additional regret that converge rapidly to zero as the number of decisions per user grows. Moreover, in the realistic setting where we know something about the relationships amongst users, we show how to obtain much tighter bounds which can be meaningful even when each user makes only a single decision. We model this information as a graph E on the set of users, perhaps a graph of relationships in a social network, and prove bounds on the additional regret based on the *fraction of honest users' friends who are dishonest*.

Although we address several fundamental questions, we leave even more open. Developing a deeper understanding of robust collaborative learning will help connect theoretical regret bounds with the practical guarantees required by applications involving many potentially dishonest users. If this project is successful, we may eventually look back on the days when learning algorithms were "secure by obscurity" in the same way that we now look back on pre-modern cryptography.

1.1 Model

Our motivating use case is an online service (such as a marketplace or ridesharing service) which must make recommendations to a large number of users. Users in turn provide feedback on the quality of those recommendations. Intuitively we expect that some users report feedback "honestly," such that a lower bound on their *reported* happiness corresponds to an actual lower bound on the performance of our algorithm. Other users may behave arbitrarily, and in particular they may adjust their feedback in order to manipulate the behavior of our algorithm.

Formally, we are given a set $\mathcal{U}$ of "users," a set $\mathcal{X}$ of "experts" or policies, and a set $\mathcal{A}$ of "arms" or actions. In each *round* $t = 0, 1, 2 \ldots$, we are given a user $u_t \in \mathcal{U}$ and a recommended distribution over actions $q_t^x \in \Delta(\mathcal{A})$ for each expert $x \in \mathcal{X}$. Before we make a decision, nature fixes a loss function $\ell_t : \mathcal{A} \to \mathbb{R}$. We select an action $a_t \in \mathcal{A}$, and then observe the loss $\ell_t(a_t)$; in prediction with expert advice we also observe the losses $\ell_t(a)$ for $a \neq a_t$, while in our other problems we do not.

In an application to online marketplaces, round t could begin when user u_t searches for a product and identifies several merchants selling that product. The actions $a \in \mathcal{A}$ correspond to merchants who we could recommend, and the experts $x \in \mathcal{X}$ correspond to rules for selecting a merchant. After recommending a merchant a_t, we observe the user's loss $\ell_t(a_t)$ from interacting with that merchant.

What kind of performance guarantee would convince us that a protocol is robust to manipulation?

Because a manipulative user may report any loss that they want, bounding the *total* loss $\sum_t \ell_t(a_t)$ is not especially helpful: a high loss by the honest users might be more than offset by a low loss from the manipulative users.

Instead, we suppose that there is some *unknown* set of users $H \subset \mathcal{U}$ whose performance we care about. Rather than consider the total loss of all users, we will consider the loss of users in H. That is, we define:

$$\ell_{<T}(H) = \sum_{\substack{t<T \\ u_t \in H}} \ell_t(a_t).$$

We will compare this to the loss $\mathrm{OPT}_{<T}(H)$ that *would have been obtained* if the users in H had chosen the best single expert $x^* \in \mathcal{X}$ and had followed their recommendation in every round:

$$\mathrm{OPT}_{<T}(H) = \min_{x^* \in \mathcal{X}} \sum_{\substack{t<T \\ u_t \in H}} \mathbb{E}_{a \sim q_t^{x^*}} \left[\ell_t(a)\right]$$

We will prove bounds on the *regret* $\ell_{<T}(H) - \mathrm{OPT}_{<T}(H)$.

In the simplest setting, the average additional regret per user will be roughly $\mathcal{O}\left(\sqrt{k\alpha}\right)$, where α is the fraction of manipulative users and k is the number of decisions per user. This bound grows sublinearly and so eventually becomes negligible, but it doesn't offer much protection until each user has made at least a handful of decisions.

Our most promising results are in the setting where we have a graph E on the set of users, such as the graph of relationships in a social network. We can then define the quantity $\alpha_E(H)$ as the probability that a randomly chosen edge of E with one endpoint in H has the other endpoint out of H, which will generally be radically smaller than α (compare the fraction of internet users who behave maliciously to the fraction of your Facebook friends who behave maliciously). In this setting, we prove per-user regret bounds that depend on $\mathcal{O}\left(\sqrt{k\alpha_E(H)}\right)$, a bound which is already meaningful when $k = 1$.

Our bounds hold for every subset H simultaneously; there is no intrinsic notion of "honesty" nor any underlying statistical model. As a result, H could consist of the "honest" users of the system, or it could consist of a subset of users who happen to have sufficiently similar tastes.

1.2 Techniques

Semidefinite programming relaxations

Our first obstacle is that all of our results require learning over the space of *partitions* of the users into clusters. Rather than working with the exponentially large space of all partitions we use a semidefinite programming relaxation, following a long tradition in approximation algorithms [15, 22]. We consider the space $\mathcal{C}$ of positive semidefinite matrices X with 1's on the diagonal and rows and columns indexed by $\mathcal{U}$. We can view such a matrix X as a "pseudodistribution" over partitions, with the entry X_{uv} indicating the probability that u and v belong to the same part of the partition.

In order to perform online learning over $\mathcal{C}$, we need to define regularizers which are sufficiently strongly convex and yet take on a sufficiently narrow range of possible values. If we were working with the space of distributions over all possible partitions then entropy regularization would be the most natural approach. It is not clear how to generalize this regularizer to a pseudodistribution $X \in \mathcal{C}$, since such a matrix need not correspond to any actual distribution over partitions. However, there is always a set of *continuous* random variables which are consistent with X in a suitable sense, and we can compute the *maximal* entropy of any such distribution—in fact, it is simply the log determinant of X. We show that the log determinant inherits the desirable properties of entropy regularization, and allows us to perform effective online learning over $\mathcal{C}$.

We next consider the case where we have prior information about which users are related, in the form of a k-regular graph $E \subset \mathcal{U} \times \mathcal{U}$, and would like to more quickly learn partitions which are consistent with these relationships. In this setting we use the modified regularizer:

$$R(X) = -\log\det(X + I) - \frac{1}{k}\operatorname{Tr}(A_E X)$$

where A_E is the adjacency matrix of E. Using this regularizer, we show how to obtain regret that depends on the number of edges crossing the partition. The key challenge is showing that R never takes on very small values, i.e. that whenever $\log\det(X + I)$ is large, $\operatorname{Tr}(A_E X)$ is small.

Prediction with expert advice

We next consider a collaborative version of prediction with expert advice. We would like to aggregate data from all of the users and then apply a traditional algorithm like multiplicative weights. The problem with this plan is that manipulative users may report high losses for good experts and low losses for bad experts.

Ideally, each expert x could simply decline to participate in any round involving a dishonest user, so that dishonest users cannot do any harm. This intuition can be formalized in the sleeping experts model, in which each expert is awake during a subset of the rounds and the learner performs nearly as well as each expert *during the subset of rounds when that expert is awake.*

The key challenge is determining when each expert x should be awake. Similar problems have been addressed in the past by passing to an exponentially large space of experts [8, 18, 1]: for each set $H \subset \mathcal{U}$, we could introduce a copy of x which is awake precisely during rounds involving a user in H. Because the regret of multiplicative weights depends logarithmically on the number of experts, this yields a statistically efficient algorithm. But because the set of experts is large, the naive algorithm is computationally intractable. The algebraic structure of multiplicative updates makes it possible to simulate this algorithm efficiently [18], but this approach requires the weights to form a product distribution over sets $H \subset \mathcal{U}$, i.e. requires us to treat each user independently. This prevents us from leveraging information about the relationships amongst users, which may be critical to effectively preventing manipulation.

Our key insight is that we can use online learning to decide when each expert x should be awake. We instantiate an independent online learning algorithm for each expert, and in round t we have that learning algorithm output a quantity $z_t(x) \in [0, 1]$ indicating the probability that expert x should be awake in round t. Ideally we could take $z_t(x) = 1$ if and only if $u_t \in H$. We approximately accomplish the same goal by choosing when expert x is awake in order to maximize the rate at which its weight increases. Then the regret bound for our learning algorithm ensures that adversarial users cannot do too much harm, since the expert will learn to set $z_t(x) = 0$ when the user is manipulative. The choice of z_t is informed by our experiences in all previous rounds as well as the relationships amongst users.

In order to compute the values z_t, we use the strategy from the previous section to compute a positive semidefinite matrix $X_t(x)$ for each expert x, defining $z_t(x)$ to be an appropriate entry of that matrix and minimizing the loss $z_t(x)\big(\ell_t(x) - \overline{\ell_t}\big)$, where $\overline{\ell_t}$ is the average loss of the awake experts. Note that we are using a different loss for each expert, and so the matrices $X_t(x)$ will quickly diverge.

This procedure guarantees that expert x outperforms the learner *while it is awake* almost as much as if it had been awake precisely in rounds involving users in H. But we know that no expert can consistently outperform the learner while it is awake. Thus no expert can consistently outperform the learner on the set of rounds involving a user in H, for any set H.

Contextual bandits

We extend these results to the contextual bandits problem, a generalization of prediction with expert advice in which only partial information about losses is available. It is straightforward to construct an unbiased estimator for the loss of each expert, but these estimators introduce considerable additional variance which increases the regret. We explicitly provide a compensating adjustment to each expert's loss that offsets the additional regret introduced by this variance. We can then apply the same strategy described in the last section, but with a more careful and involved analysis.

Collaborative filtering

Finally, we consider a setting where users must decide which resources they are willing to interact with: a filtering strategy effectively corresponds to a matrix with rows indexed by the set of users $\mathcal{U}$ and columns indexed by the set of resources $\mathcal{I}$. Past work has used matrix prediction directly to compute such a matrix, but we show that this approach is vulnerable to manipulation. Instead, we view a partition $\mathcal{U} \cup \mathcal{I} = A \cup B \cup C \cup D$ as a *modification* to a filtering strategy: users in A should start interacting with resources in B, while users in C should stop interacting with resources in D. A distribution over partitions then corresponds to a stochastic modification to a filtering strategy, and we use the filtering strategy which is a fixed point of this distribution. (This is analogous to the use of a fixed point computation in [8].) If we choose our partitions to maximize the improvement from the resulting update, we can conclude that there exists *no* update which substantially improves the total payoff of all users, and in particular that every group of users H must be making approximately optimal decisions.

1.3 Related work

Online learning with time selection functions. Optimizing the welfare of an unknown group of users H is a special case of the model in [8], where the learner is given a set of "time selection functions," each of which assigns a weight to each timestep, and wants to have low regret for *all* of these weightings. In our setting, we must deal with an exponentially large family of possible time selection functions, one for every subset $H \subset \mathcal{U}$, and so applying the generic algorithm from [8] incurs an exponential computational overhead. Instead, we provide algorithms which exploit the special structure of the collaborative learning problem in order to remain computationally efficient.

Multitask online learning. Our setup is analogous to the model of multi-task prediction introduced in [1] and studied in [18], and in particular to their *shifting multitask problem*. In their setting, one "task" is active in each round, analogous to our user. Our regret bounds are strictly stronger than the multi-task regret bounds they study, although we can adapt the technique from [18] in order to achieve a manipulation-resistant algorithm for prediction with expert advice. We go substantially beyond existing work by considering a broader range of learning problems and by showing how to leverage information about relationships amongst users. Our techniques could also be applied in the multi-task setting, though our model of relationships and collaborative filtering are especially relevant in applications involving multiple users.

Competitive collaborative learning. Awerbuch and Kleinberg propose a similar model of collaborative learning [7], and study the multi-armed bandit problem (a special case of contextual bandits) in this setting. The difference is that their model rests on statistical assumptions: that each honest user would obtain the same (expected) loss from each resource at each point in time. This is a strong assumption which is likely to be violated if some of the resources behave adversarially. If we specialize our contextual bandits algorithm to the multi-armed bandit problem we obtain weaker regret bounds than [7], but in Chapter 4 we show that the stronger bounds cannot be achieved without statistical assumptions and that our results are optimal in the worst case.

Collaborative filtering: In the collaborative filtering problem, a set of users interact with a set of resources, and exploit their common tastes to more efficiently predict which resources each of them will rate highly. This problem has been studied at length; see [29] for an overview.

In contrast with this literature, we focus on robustness and non-manipulability. Existing work makes very weak guarantees if we include even a small fraction of users who behave manipulatively. Most existing work also makes strong assumptions on the relationship between the preferences of different users (such as the existence of an approximately low-rank decomposition), while we only assume that there exists a set of users who could benefit by pooling their information.

Collaborative preference learning: Some collaborative filtering systems, such as [2, 6] make guarantees for every set of users and are robust to adversarial manipulation. However, these results make strong assumptions—that preferences are static over time and approximately shared by many users—and assume that users are free to choose what resources to interact with. In contrast, we make minimal assumptions, unconditionally competing with the best fixed benchmark even if preferences vary arbitrarily across users and over time.

Matrix prediction: Our semidefinite programming algorithm for online local learning improves upon recent results in matrix prediction due to Hazan, Kale, and Shalev-Schwartz [16], based on von Neumann entropy regularization [3]. We use a different regularizer and analysis, and obtain the first asymptotically optimal bounds for local learning. In the collaborative learning setting, our improvement translates into per user regret which is a constant independent of the total number of users, an important qualitative improvement. Moreover, a direct application of existing matrix prediction results to collaborative filtering would not yield a manipulation-resistant algorithm.

Manipulation-resistance: Another literature attempts to modify reputation systems to limit the influence of sybils, fake identities controlled by an attacker. For example, see [30, 25]. However, these algorithms do not give any non-trivial statistical bounds in any of the settings we consider.

Chapter 2

Online convex optimization and experts

Online convex optimization is a foundational problem in statistical learning theory. Several of our collaborative learning problems will be special cases of online convex optimization. Moreover, online convex optimization, especially the special case prediction with expert advice, frequently arises as a subroutine in our algorithms.

Online mirror descent (OMD) is a successful algorithm for online convex optimization. We will need to make extensive use of OMD in later chapters, and so we introduce the algorithm and its analysis here. We will also need a version of mirror descent with some additional advantages—especially leveraging the predictability of the reward and handling "specialists" who sometimes decline to offer advice. So we also present slightly modified algorithms that achieve these properties.

All of the analysis in this section is standard; to our knowledge this particular combination of properties has not appeared in the literature, but obtaining it does not require any new ideas.

Online convex optimization

In the online convex optimization problem, we are given a convex set $\mathcal{C} \subset \mathbb{R}^N$. At each time $t = 0, 1, 2, \ldots$ we must pick an element $x_t \in \mathcal{C}$, and then we observe a convex loss function $\ell_t : \mathcal{C} \to \mathbb{R}$. Our goal is to minimize the total loss $\sum_t \ell_t(x_t)$, and in particular to achieve a loss not much higher than $\min_{x^* \in \mathcal{C}} \sum_{t<T} \ell_t(x^*)$.

2.1 Online mirror descent

Our presentation and analysis of online mirror descent (OMD) closely follows [26].

Our first observation is that we can essentially assume that the losses ℓ_t are linear. That is, for any convex loss function ℓ_t and any $x_t \in \mathcal{C}$, there exists a subgradient g_t such that for every $x' \in \mathcal{C}$:

$$\ell_t(x') \geq \ell_t(x_t) + g_t \cdot (x' - x_t).$$

If ℓ_t is differentiable, then $g_t = \nabla \ell_t(x)$. For general convex functions, there may be many subgradients at a point. For example, any $g \in [-1, 1]$ is a subgradient of $|x|$ at the point $x = 0$.

If g_t is a subgradient of ℓ_t at x_t, then we have:

$$\sum_{t<T} (\ell_t(x_t) - \ell_t(x^*)) \leq \sum_{t<T} (g_t \cdot x_t - g_t \cdot x^*).$$

Thus a bound on the regret for the linear loss $g_t \cdot x$ immediately implies a bound on the regret for the convex loss ℓ_t. Henceforth, we will take g_t to be some subgradient of ℓ_t at x_t, and work exclusively with the subgradients g_t.

For convenience, write $g_{<t} = \sum_{t'<t} g_{t'}$.

The online mirror descent algorithm is parameterized by a regularizer $R : \mathcal{C} \to \mathbb{R}$. In round t, we choose the output that minimizes the retrospective loss $g_{<t} \cdot x$, plus the regularization $R(x)$. That is, we take

$$x_t = \operatorname*{arg\,min}_{x \in \mathcal{C}} \left(g_{<t} \cdot x + R(x) \right).$$

This is the entire OMD algorithm.

Analysis of mirror descent

We will analyze OMD by using the Frenchel conjugate of the regularizer R as a potential function. The Frenchel conjugate R^* is defined as:

$$R^*(g) = \max_{x \in \mathcal{C}} \left(g \cdot x - R(x) \right).$$

(This is actually the Frenchel conjugate of the function which is equal to R on $\mathcal{C}$ and equal to $+\infty$ everywhere else. Throughout this chapter we will write R^* for the conjugate of this modified function, with the set $\mathcal{C}$ typically clear from context.)

It's easy to see that $\nabla R^*(g)$ is precisely the $x \in \mathcal{C}$ for which $(g \cdot x - R(x))$ is maximal. In particular, mirror descent outputs $x_t = \nabla R^*(-g_{<t})$.

Our notation will be simplified by the concept of a Bregman divergence, which we will use to measure how quickly our regularizer R changes. For a convex function R and two inputs x, x', we define

$$D_R\left(x \parallel x'\right) = R(x) - R(x') - (x - x') \cdot \nabla R(x').$$

In words, the Bregman divergence between x and x' is the difference between $R(x)$ and the first-order approximation to $R(x)$ centered at x'. For example, the Bregman divergence associated with the squared Euclidean norm is the squared Euclidean distance. The Bregman divergence associated with entropy is the KL divergence.

The following theorem is our key tool for analyzing OMD:

Lemma 1 ([26] Lemma 2.20)**.** *Fix a convex set $\mathcal{C}$ and a convex $R : \mathcal{C} \to \mathbb{R}$. For any $T > 0$, $g_t \in \mathbb{R}^N$, and $x^* \in \mathcal{C}$, OMD over $\mathcal{C}$ with losses g_t and convex regularizer R satisfies:*

$$\sum_{t<T} g_t \cdot x_t \leq \sum_{t<T} g_t \cdot x^* + \sum_{t<T} D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) + \left(R(x^*) - \min_{x \in \mathcal{C}} R(x)\right)$$

Proof. We have:

$$\begin{aligned} R^*(-g_{<t+1}) - R^*(-g_{<t}) &= (g_{<t} - g_{<t+1}) \cdot \nabla R^*(-g_t) + D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) \\ &= -g_t \cdot x_t + D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) \end{aligned}$$

Also, $R^*(g) \geq -g \cdot x^* - R(x^*)$ for any $g \in \mathbb{R}^N, x^* \in \mathcal{C}$, and $R^*(0) = -\min_{x \in \mathcal{C}} R(x)$. Putting this all together:

$$\begin{aligned} g_{<T} \cdot x^* &\geq -R(x^*) - R^*(-g_{<T}) \\ &= -R(x^*) - R^*(-g_{<0}) - \sum_{t<T}(R^*(-g_{<t+1}) - R^*(-g_{<t})) \\ &= -R(x^*) - R^*(0) + \sum_{t<T} g_t \cdot x_t - \sum_{t<T} D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) \\ \sum_{t<T} g_t \cdot x_t &\leq g_{<T} \cdot x^* + \sum_{t<T} D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) + \left(R(x^*) - \min_{x \in \mathcal{C}} R(x)\right) \end{aligned}$$

as desired. □

.

OMD with strictly convex regularizers

We say that a function R is β-strongly-smooth with respect to a norm $\|\cdot\|$ if:

$$D_R\left(x \parallel x'\right) \leq \frac{\beta}{2} \left\|x - x'\right\|^2.$$

If R^* is strongly smooth with respect to a norm $\|\cdot\|$, we immediately obtain a regret bound for OMD in term of $\sum_t \|g_t\|^2$.

Similarly, we say that R is β-strongly-convex if:

$$D_R\left(x \parallel x'\right) \geq \frac{\beta}{2} \left\|x - x'\right\|^2.$$

Strong convexity is relevant for our purposes essentially because it is "dual" to strong smoothness.

That is, given a norm $\|\cdot\|$, we define the *dual norm* $\|\cdot\|_\star$ as

$$\|x\|_\star = \max_{y : \|y\| \leq 1} y \cdot x.$$

Then we have:

Lemma 2 ([26] Lemma 2.19)**.** *If R is β-strongly-convex with respect to a norm $\|\cdot\|$, then R^* is β^{-1}-strongly-smooth with respect to the dual norm $\|\cdot\|_\star$.*

Proof. Pick g, g' arbitrarily, and let $x' = \arg\max_{x\in\mathcal{C}}(g' \cdot x - R(x)) = \nabla R^*(g')$. We have

$$\begin{aligned}
D_{R^*}(g \parallel g') &= R^*(g) - R^*(g') - (g - g') \cdot \nabla R^*(g') \\
&= \max_{x\in\mathcal{C}}(g \cdot x - R(x)) - g' \cdot x' + R(x') - (g - g') \cdot x' \\
&= \max_{x\in\mathcal{C}}(g \cdot x - R(x)) + R(x') - g \cdot x' \\
&= \max_{x\in\mathcal{C}}(g \cdot x - R(x') - g' \cdot (x - x') - D_R(x \parallel x')) + R(x') - g \cdot x' \\
&= \max_{x\in\mathcal{C}}((g - g') \cdot (x - x') - D_R(x \parallel x')) \\
&\le \max_{x\in\mathcal{C}}\left((g - g') \cdot (x - x') - \frac{\beta}{2}\|x - x'\|^2\right) \\
&\le \max_{x\in\mathcal{C}}\left(\|g - g'\|_\star \|x - x'\| - \frac{\beta}{2}\|x - x'\|^2\right) \\
&\le \max_{d\in\mathbb{R}}\left(\|g - g'\|_\star d - \frac{\beta}{2}d^2\right) \\
&= \frac{1}{2\beta}\|g - g'\|_\star^2
\end{aligned}$$

as desired. □

Combining this with Lemma 1, we obtain:

Theorem 1 ([26])**.** *Fix a convex set $\mathcal{C}$ and a function $R : \mathcal{C} \to \mathbb{R}$ which is β-strongly convex with respect to the norm $\|\cdot\|$. Let $\|\cdot\|_\star$ be the dual norm to $\|\cdot\|$. For any $T > 0$, $g_t \in \mathbb{R}^N$, $\eta > 0$, and $x^* \in \mathcal{C}$, OMD over $\mathcal{C}$ with losses g_t and regularizer $\eta^{-1}R$ satisfies:*

$$\sum_{t<T} g_t \cdot x_t \le \sum_{t<T} g_t \cdot x^* + \eta\beta^{-1} \sum_{t<T} \|g_t\|_\star^2 + \eta^{-1}\left(R(x^*) - \min_{x\in\mathcal{C}} R(x)\right).$$

Proof. Note that $\eta^{-1}R$ is $\eta^{-1}\beta$-strongly convex with respect to $\|\cdot\|$. By by Lemma 2, $(\eta^{-1}R)^*$ is $\eta\beta^{-1}$-strongly smooth with respect to $\|\cdot\|_\star$. Then Lemma 1 directly yields the claimed bound. □

Mirror descent over the simplex

We will often be interested in prediction with expert advice, and in particular the decision-theoretic or Hedge setting [14]. In this case we are given a finite set $\mathcal{X}$ of "experts," and take

$$\mathcal{C} = \Delta(\mathcal{X}) = \left\{ p : \mathcal{X} \to [0, 1] \;\middle|\; \sum_{x\in\mathcal{X}} p[x] = 1 \right\}.$$

We view the gradient $g[x]$ as the loss of expert x, and $p \in \mathcal{C}$ as a stochastic choice of expert. The minimum of $g_{<T} \cdot p$ will always be obtained at a p supported on a single expert. So our task is to compete with the loss of the best fixed expert.

The negative entropy

$$R_{\mathrm{H}}(p) = \sum_{x \in \mathcal{X}} p[x] \log p[x]$$

is a common regularizer for the probability simplex. This regularizer takes values between 0 and $\log |\mathcal{X}|$. The conjugate is

$$\left(\eta^{-1} R_{\mathrm{H}}\right)^*(g) = \frac{1}{\eta} \log\left(\sum_{x \in \mathcal{X}} \exp(\eta g[x])\right).$$

We can directly compute and bound the Bregman divergences of R^*. As before, set $g_{<t} = \sum_{t'<t} g_{t'}$, and $p_t = \nabla R^*(g_{<t})$.

Lemma 3 ([26] Theorem 2.22)**.** *Fix any $\eta > 0$ and any $a, b : \mathcal{X} \to \mathbb{R}$ satisfying $\eta(a[x] - b[x]) \geq -1$. Let $p = \nabla(\eta^{-1} R_{\mathrm{H}})^*(b)$. Then:*

$$D_{(\eta^{-1} R_{\mathrm{H}})^*}\left(a \parallel b\right) \leq \eta \sum_{x \in \mathcal{X}} p[x] (a[x] - b[x])^2$$

Combining this with Lemma 1, we obtain:

Theorem 2 ([26] Theorem 2.22)**.** *For any $x^* \in \mathcal{X}$, $T > 0$, $\eta > 0$, and $g_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta g_t[x] \geq -1$, OMD over the probability simplex with regularizer $\eta^{-1} R_{\mathrm{H}}$ and losses g_t satisfies:*

$$\sum_{t<T} g_t \cdot p_t \leq \sum_{t<T} g_t[x^*] + \eta \sum_{x \in \mathcal{X}} p_t[x] g_t[x]^2 + \eta^{-1} \log |\mathcal{X}|.$$

2.2 Prediction with expert advice

Non-uniform prior

In some cases we have a prior belief about which experts are likely to perform well, and may prefer to guarantee lower regret against the most promising experts (at the expense of higher regret against less promising experts). This can be achieved easily in the mirror descent framework.

Suppose that we have a prior distribution $w \in \Delta(\mathcal{X})$. Rather than using the entropy R_{H} as our regularizer, we can use the KL divergence $D_{R_{\mathrm{H}}}\left(\cdot \parallel w\right)$. The KL divergence is non-negative. Moreover, because the KL divergence differs from the entropy by a linear function, its dual $D_{R_{\mathrm{H}}}\left(\cdot \parallel w\right)^*$ is a translated version of R_{H}^*, and hence satisfies Lemma 3. Thus we obtain

Theorem 3. *For any $x^* \in \mathcal{X}$, $w \in \Delta(\mathcal{X})$, $T > 0$, $\eta > 0$, and $g_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta g_t[x] \geq -1$, OMD over the probability simplex with regularizer $D_{\eta^{-1}R_{\mathrm{H}}}(\cdot \parallel w)$ and losses g_t satisfies*

$$\sum_{t<T} g_t \cdot p_t \leq \sum_{t<T} g_t[x^*] + \eta \sum_{x \in \mathcal{X}} p_t[x] g_t[x]^2 + \eta^{-1} \log \frac{1}{w[x^*]}$$

Optimism

Suppose that before choosing p_t we have an estimate μ_t for g_t. If this estimate were exactly accurate, then we could simply choose p_t to optimize μ_t, and so experience no regret. Intuitively, this suggests that we might be able to obtain a regret bound that depends on our error $g_t - \mu_t$ rather than on the losses g_t themselves. Such bounds are established by [24], who introduced the model of learning with predictable loss sequences—we reproduce them here for completeness.

For online convex optimization in general, we make the straightforward modification of outputting

$$x_t = \nabla R^*(-g_{<t} - \mu_t),$$

i.e. including μ_t in the optimization alongside the losses $g_{<t}$ from previous rounds. We can then improve Lemma 1 by a simple change to the analysis:

Lemma 4. *Fix a convex set $\mathcal{C}$ and a convex regularizer $R : \mathcal{C} \to \mathbb{R}$. For any $T > 0$, $g_t, \mu_t \in \mathbb{R}^N$, and $x^* \in \mathcal{C}$, optimistic OMD over $\mathcal{C}$ with regularizer R, losses g_t, and predicted losses μ_t satisfies:*

$$\sum_{t<T} g_t \cdot x_t \leq \sum_{t<T} g_t \cdot x^* + \sum_{t<T} D_{R^*}(-g_{<t+1} \parallel -g_{<t} - \mu_t) + \left(R(x^*) - \min_{x \in \mathcal{C}} R(x) \right)$$

Proof. Note that R^* is convex. Thus for any a, b we have $R^*(a) - R^*(b) \leq (a - b) \cdot \nabla R^*(a)$. Our modified algorithm satisfies a tighter bound on $R^*(-g_{<t+1}) - R^*(-g_{<t})$:

$$\begin{aligned}
R^*(-g_{<t+1}) - R^*(-g_{<t}) &= (R^*(-g_{<t+1}) - R^*(-g_{<t} - \mu_t)) + (R^*(-g_{<t} - \mu_t) - R^*(-g_{<t})) \\
&= (\mu_t - g_t) \cdot x_t + D_{R^*}(-g_{<t+1} \parallel -g_{<t} - \mu_t) \\
&\quad + (R^*(-g_{<t} - \mu_t) - R^*(-g_{<t})) \\
&\leq (\mu_t - g_t) \cdot x_t + D_{R^*}(-g_{<t+1} \parallel -g_{<t} - \mu_t) - \mu_t \cdot \nabla R^*(-g_{<t} - \mu_t) \\
&= (\mu_t - g_t) \cdot x_t + D_{R^*}(-g_{<t+1} \parallel -g_{<t} - \mu_t) - \mu_t \cdot x \\
&= -g_t \cdot x_t + D_{R^*}(-g_{<t+1} \parallel -g_{<t} - \mu_t)
\end{aligned}$$

Then, as in Lemma 1:

$$\begin{aligned}
g_{<T} \cdot x^* &\geq -R^*(-g_{<T}) - R(x^*) \\
&= -R^*(-g_{<0}) - R(x^*) - \sum_{t<T} (R^*(-g_{<t+1}) - R^*(-g_{<t})) \\
&\geq R(x_0) - R(x^*) + \sum_{t<T} g_t \cdot x_t - \sum_{t<T} D_{R^*}\left(-g_{<t+1} \parallel -g_{<t} - \mu_t\right) \\
\sum_{t<T} g_t \cdot x_t &\leq g_{<T} \cdot x^* + \sum_{t<T} D_{R^*}\left(-g_{<t+1} \parallel -g_{<t}\right) + \left(R(x^*) - \min_{x \in \mathcal{C}} R(x)\right)
\end{aligned}$$

as desired. □

By applying this lemma in combination with the KL divergence regularizer and Lemma 3, we obtain:

Theorem 4. *For any $x^* \in \mathcal{X}$, $w \in \Delta(\mathcal{X})$, $T > 0$, $\eta > 0$, and $g_t, \mu_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta(g_t[x] - \mu_t[x]) \geq -1$, OMD over $\Delta(\mathcal{X})$ with regularizer $D_{\eta^{-1}R_{\mathrm{H}}}(\cdot \parallel w)$, losses g_t, predicted losses μ_t, and prior w satisfies*

$$\sum_{t<T} g_t \cdot p_t \leq \sum_{t<T} g_t[x^*] + \eta \sum_{x \in \mathcal{X}} p_t[x](g_t[x] - \mu_t[x])^2 + \eta^{-1} \log \frac{1}{w[x^*]}$$

Learning from specialists

We will often be interested in the so-called "specialists" or "sleeping experts" setting, in which each expert $x \in \mathcal{X}$ is awake only on some subset of rounds, and our goal is to compete with each expert on the set of rounds where that expert is awake.

More precisely: in each round (prior to choosing a distribution over experts) we are given a map $z_t : \mathcal{X} \to [0, 1]$. Write $\widehat{p_t} \in \Delta(\mathcal{X})$ for our selection in round t. Write $\widehat{g_t} : \mathcal{X} \to \mathbb{R}$ for the losses incurred by the experts in round t, and $\overline{g_t} = \widehat{p_t} \cdot \widehat{g_t}$ for the loss incurred by our algorithm in round t. Write $\widehat{\mu_t} : \mathcal{X} \to \mathbb{R}$ for the predicted losses in round t, and $\overline{\mu_t} = \widehat{p_t} \cdot \widehat{\mu_t}$ for the predicted loss incurred by our algorithm in round t.

Our goal is to bound the regret compared to expert x^* in those rounds when x^* is awake:

$$\sum_{t<T} z_t[x^*](\overline{g_t} - \widehat{g_t}[x^*])$$

These guarantees can be obtained by a simple modification of our algorithm. We define synthetic losses

$$g_t[x] = z_t[x]\widehat{g_t}[x] + (1 - z_t[x])\overline{g_t},$$

and synthetic predicted losses

$$\mu_t[x] = z_t[x]\widehat{\mu_t}[x] + (1 - z_t[x])\overline{\mu_t}.$$

We then use optimistic OMD to produce a distribution p_t, and define:

$$\widehat{p_t}[x] = \frac{p_t[x] z_t[x]}{\sum_{y \in \mathcal{X}} p_t[y] z_t[y]}.$$

This definition is slightly incomplete, because μ_t depends on $\overline{\mu_t}$ which depends on p_t (which itself depends on μ_t). However, if we make a guess m_t for $\overline{\mu_t}$, we can use use that to compute provisional values $\mu_t'(m_t)$, then $p_t'(m_t)$, then $\overline{\mu_t}'(m_t)$. If $\overline{\mu_t}'(m_t) \approx m_t$, then we can use m_t in place of $\overline{\mu_t}$ in the definition of the μ_t, and still approximately satisfy the defining equation relating μ_t to $\overline{\mu_t}$. In fact it is easy to find an m_t such that this approximation is very good.

Observe that $\overline{\mu_t}'(m_t)$ is always between the lowest and highest coordinate of $\widehat{\mu_t}$. Thus $m_t - \overline{\mu_t}'(m_t)$ is negative for small values of m_t and positive for large values of m_t. By performing a binary search we can find an exponentially small interval in which $m_t - \overline{\mu_t}'(m_t)$ changes sign. It's easy to calculate that $\overline{\mu_t}'(m_t)$ is a Lipschitz function of m_t, with Lipschitz constant 4 (though even an exponential Lipschitz constant would be sufficient for our purposes). Thus the value of $m_t - \overline{\mu_t}'(m_t)$ is exponentially small on this interval. Thus we can compute values of m_t such that the equation above is satisfied up to some exponentially small tolerance.

Algorithm 1: EXPERTS$_{\text{sleep}}$

```
function Init_sleep($(\mathcal{X}, w, \eta)$)
    $g_{<0} \leftarrow 0$;
    Return $S_0 = (\mathcal{X}, w, \eta, g_{<0})$;
function Predict_sleep($(S_t = (\mathcal{X}, w, \eta, g_{<t}), z_t, \widehat{\mu_t})$)
    function $\widehat{p}(\overline{\mu})$
        $\mu[x] \leftarrow z_t[x]\widehat{\mu_t}[x] + (1 - z_t[x])\overline{\mu}$;
        $p \leftarrow \min_{p \in \Delta(\mathcal{X})} (g_{<t} + \mu_t) \cdot p + D_{\eta^{-1} R_{\mathrm{H}}}(p \parallel w)$;
        Return $\widehat{p}[x] = \frac{z[x]p[x]}{\sum_y z[y]p[y]}$;
    $\overline{\mu_t} \leftarrow$ search for $\overline{\mu}$ such that $\overline{\mu} \approx \widehat{p}(\overline{\mu}) \cdot \widehat{\mu_t}$ ;
    Return $\widehat{p_t} = \widehat{p}(\overline{\mu_t})$;
function Update_sleep($(S_t = (\mathcal{X}, w, \eta, g_{<t}), z_t, \widehat{\mu_t}, \widehat{g_t})$)
    $\widehat{p_t} \leftarrow$ Predict_sleep$(S_t, z_t, \widehat{\mu_t})$;
    $\overline{g_t} \leftarrow \widehat{g_t} \cdot \widehat{p_t}$;
    $g_t[x] \leftarrow z_t[x]\widehat{g_t}[x] + (1 - z_t[x])\overline{g_t}$;
    $g_{<t+1} \leftarrow g_{<t} + g_t$;
    Return $S_{t+1} = (\mathcal{X}, w, \eta, g_{<t+1})$;
```

We can now define the optimistic specialists algorithm. Given the predicted losses $\widehat{\mu_t}$ we compute the estimate m_t as described in the previous paragraph. We then use this value to compute μ_t, p_t, and $\widehat{p_t}$. We play the distribution $\widehat{p_t}$, and then report the losses g_t as defined above.

Lemma 5. *Fix any $w \in \Delta(\mathcal{X})$, $T > 0, \eta > 0$, $z_t : \mathcal{X} \to [0,1]$ and $\widehat{g_t}, \widehat{\mu_t} : \mathcal{X} \to \mathbb{R}$ satisfying $\eta(\widehat{g_t}[x] - \widehat{\mu_t}[x]) \geq -1$. Set $S_0 = \mathrm{Init}_{\mathrm{sleep}}(\mathcal{X}, w, \eta)$, $\widehat{p_t} = \mathrm{Predict}_{\mathrm{sleep}}(S_t, z_t, \widehat{\mu_t})$, and $S_{t+1} = \mathrm{Update}_{\mathrm{sleep}}(S_t, z_t, \widehat{\mu_t}, \widehat{g_t})$. Then for every $x^* \in \mathcal{X}$:*

$$\sum_{t<T} z_t[x^*]\widehat{g_t} \cdot \widehat{p_t} \leq \sum_{t<T} z_t[x^*]\widehat{g_t}[x^*] + 4\eta \sum_{x \in \mathcal{X}} \widehat{p_t}[x](\widehat{g_t}[x] - \widehat{\mu_t}[x])^2 + \eta^{-1} \log \frac{1}{w[x^*]}$$

Proof. We will apply Theorem 4. First we evaluate each of its expression separately, by plugging in our definitions of $g_t, \mu_t, \widehat{p_t}$. Write $W_t = \sum_{x \in \mathcal{X}} p_t[x] z_t[x]$.

$$\begin{aligned}
\sum_{t<T} g_t \cdot p_t &= \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x](z_t[x]\widehat{g_t}[x] + (1 - z_t[x])\overline{g_t}) \\
&= \sum_{t<T} \left(W_t \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\widehat{g_t}[x] + \overline{g_t} - \sum_{x \in \mathcal{X}} p_t[x] z_t[x] \overline{g_t} \right) \\
&= \sum_{t<T} (W_t \overline{g_t} + \overline{g_t} - W_t \overline{g_t}) \\
&= \sum_{t<T} \overline{g_t}
\end{aligned}$$

Thus

$$\begin{aligned}
\sum_{t<T} g_t \cdot p_t - g_t[x^*] &= \sum_{t<T} \overline{g_t} - \sum_{t<T} z_t[x^*]\widehat{g_t}[x^*] - \sum_{t<T} (1 - z_t[x^*])\overline{g_t} \\
&= \sum_{t<T} z_t[x^*](\overline{g_t} - \widehat{g_t}[x^*]).
\end{aligned}$$

We now turn our attention to the second order term. Write $\varepsilon_t[x] = \widehat{g_t}[x] - \widehat{\mu_t}[x]$ and $\overline{\varepsilon_t} = \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\varepsilon_t[x]$.

$$\begin{aligned}
\sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x](g_t[x] - \mu_t[x])^2 &\approx \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x](z_t[x]\varepsilon_t[x] + (1 - z_t[x])\overline{\varepsilon_t})^2 \\
&\leq 2 \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x] z_t[x]^2 \varepsilon_t[x]^2 + 2 \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x](1 - z_t[x])^2 \overline{\varepsilon_t}^2 \\
&\leq 2 \sum_{t<T} \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\varepsilon_t[x]^2 + 2 \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x] \left(\sum_{y \in \mathcal{X}} \widehat{p_t}[y]\varepsilon_t[y] \right)^2 \\
&= 2 \sum_{t<T} \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\varepsilon_t[x]^2 + 2 \sum_{t<T} \left(\sum_{y \in \mathcal{X}} \widehat{p_t}[y]\varepsilon_t[y] \right)^2 \\
&\leq 2 \sum_{t<T} \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\varepsilon_t[x]^2 + 2 \sum_{t<T} \left(\sum_{y \in \mathcal{X}} \widehat{p_t}[y] \right) \left(\sum_{y \in \mathcal{X}} \widehat{p_t}[y]\varepsilon_t[y]^2 \right) \\
&= 4 \sum_{t<T} \sum_{x \in \mathcal{X}} \widehat{p_t}[x]\varepsilon_t[x]^2
\end{aligned}$$

where the first equality is approximate because $\overline{\mu_t}$ was approximated by a binary search, and the final inequality holds by Cauchy-Schwartz.

Now we apply Theorem 4 and combine with these identities:

$$\begin{aligned}\sum_{t<T} g_t \cdot p_t &\leq \sum_{t<T} g_t[x^*] + \eta \sum_{x\in\mathcal{X}} p_t[x](g_t[x] - \mu_t[x])^2 + \eta^{-1} \log \frac{1}{w[x^*]} \\ \sum_{t<T} z_t[x^*]\widehat{g}_t \cdot \widehat{p}_t &\leq \sum_{t<T} z_t[x^*]g_t[x^*] + \eta \sum_{x\in\mathcal{X}} p_t[x](g_t[x] - \mu_t[x])^2 + \eta^{-1} \log \frac{1}{w[x^*]} \\ &\leq \sum_{t<T} z_t[x^*]g_t[x^*] + 4\eta \sum_{x\in\mathcal{X}} \widehat{p}_t[x]\varepsilon_t[x]^2 + \eta^{-1} \log \frac{1}{w[x^*]}\end{aligned}$$

as desired. □

Clipping gradients

When the errors $g_t[x] - \mu_t[x]$ are very large and positive, second order regret bounds that depend on $g_t[x]^2$ become very weak. We can obtain stronger bounds by clipping the gradients and performing some simple algebra.

Define $\widetilde{g}_t[x] = \min\{g_t[x], \eta^{-1}\}$. OMD on the probability simplex with clipped gradients is simply OMD with the losses $\widetilde{g}_t$ rather than g_t.

Theorem 5. *For $x^* \in \mathcal{X}$, $w \in \Delta(\mathcal{X})$, $T > 0$, $\eta > 0$, and $g_t, \mu_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta(g_t[x] - \mu_t[x]) \geq -1$, clipped OMD over the probability simplex with regularizer $D_{\eta^{-1}R_{\mathrm{H}}}(\cdot \parallel w)$, losses g_t, predicted losses μ_t, and prior w satisfies*

$$\sum_{t<T} g_t \cdot p_t \leq \sum_{t<T} g_t[x^*] + \eta \sum_{t<T} \sum_{x\in\mathcal{X}} p_t[x]g_t[x]\widetilde{g}_t[x] + \eta^{-1} \log \frac{1}{w[x^*]}$$

Proof. We can apply Theorem 3 to the sequence of losses $\widetilde{g}_t$:

$$\sum_{t<T} \widetilde{g}_t \cdot p_t \leq \sum_{t<T} \widetilde{g}_t[x^*] + \eta \sum_{t<T} \sum_{x\in\mathcal{X}} p_t[x]\widetilde{g}_t[x]^2 + \eta^{-1} \log \frac{1}{w[x^*]}$$

Note that in any coordinate where $g_t \neq \widetilde{g}_t$, we have $\widetilde{g}_t\eta = 1$. Thus $(g_t[x] - \widetilde{g}_t[x])\widetilde{g}_t[x]\eta = (g_t[x] - \widetilde{g}_t[x])$. Adding $\sum(g_t - \widetilde{g}_t) \cdot p_t$ to both sides, and then using the fact that $\widetilde{g}_t \leq g_t$, we have:

$$\begin{aligned}\sum_{t<T} g_t \cdot p_t &\leq \sum_{t<T} \widetilde{g}_t[x^*] + \eta \sum_{t<T} \sum_{x\in\mathcal{X}} p_t[x]\widetilde{g}_t[x]g_t[x] + \eta^{-1} \log \frac{1}{w[x^*]} \\ &\leq \sum_{t<T} g_t[x^*] + \eta \sum_{t<T} \sum_{x\in\mathcal{X}} p_t[x]\widetilde{g}_t[x]g_t[x] + \eta^{-1} \log \frac{1}{w[x^*]}\end{aligned}$$

□

In the optimistic specialists case, we instead use the clipped gradients

$$\widetilde{g}_t[x] = \widehat{g}_t[x] + \min\left\{\widehat{g}_t[x] - \widehat{\mu}_t[x], \eta^{-1}\right\} .$$

To state its bounds, define $\delta_\eta(g) = g \min\{g, \eta^{-1}\}$. Then an analysis identical to Theorem 5, but starting from Lemma 5, yields:

```
Algorithm 2: EXPERTS$_{\text{clip}}$
  function Init$_{\text{clip}}(\mathcal{X}, w, \eta)$
      Return $S_0 = \text{Init}_{\text{sleep}}(\mathcal{X}, w, \eta)$;
  function Predict$_{\text{clip}}(S_t, z_t, \widehat{\mu}_t)$
      Return $\widehat{p}_t = \text{Predict}_{\text{sleep}}(S_t, z_t, \widehat{\mu}_t)$;
  function Update$_{\text{clip}}(S_t = (\mathcal{X}, w, \eta, g_{<t}), z_t, \widehat{\mu}_t, \widehat{g}_t)$
      $\widetilde{g}_t[x] \leftarrow \widehat{\mu}_t[x] + \min\{\widehat{g}_t[x] - \widehat{\mu}_t[x], \eta^{-1}\}$;
      Return $S_{t+1} = \text{Update}_{\text{sleep}}(S_t, z_t, \widehat{\mu}_t, \widetilde{g}_t)$;
```

Theorem 6. *For any $w \in \Delta(\mathcal{X})$, $T > 0, \eta > 0$, $z_t : \mathcal{X} \to [0,1]$, and $\widehat{g}_t, \widehat{\mu}_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta(\widehat{g}_t[x] - \widehat{\mu}_t[x]) \geq -1$. Set $S_0 = \text{Init}_{\text{clip}}(\mathcal{X}, w, \eta)$, $\widehat{p}_t = \text{Predict}_{\text{clip}}(z_t, S_t, \widehat{\mu}_t)$, $S_{t+1} = \text{Update}_{\text{clip}}(S_t, z_t, \widehat{\mu}_t, \widehat{g}_t)$. Then for every $x^* \in \mathcal{X}$:*

$$\sum_{t<T} \widehat{g}_t \cdot \widehat{p}_t \leq \sum_{t<T} \widehat{g}_t[x^*] + \eta \sum_{t<T} \sum_{x \in \mathcal{X}} p_t[x] \delta_\eta(\widehat{g}_t[x] - \widehat{\mu}_t[x]) + \eta^{-1} \log \frac{1}{w[x^*]}$$

Adaptivity

Our regret bounds have so far depended on the quantity $\eta \sum_{x \in \mathcal{X}} p_t[x](g_t[x] - \mu_t[x])^2$. In recent work, [28] introduce the adaptive exponentiated gradient algorithm, which obtains a bound of the form $\eta(g_t[x^*] - \mu_t[x^*])^2$, where x^* is the expert with which we are competing. If there are any experts who have small or highly predictable losses, this bound may be much better. It is easy to verify that the modified version continues to work with a non-uniform prior w (apply Corollary 3.2 from [28] to the KL divergence regularizer, with the analysis proceeding exactly as in Corollary 3.3). A more careful proof of Corollary 3.3 also establishes the same results under the condition $\eta \left|g_t[x] - \mu_t[x]\right| \leq 1/4$.

Theorem 7 ([28], essentially equivalent to Corollary 3.3)**.** *For any $x^* \in \mathcal{X}$, $w \in \Delta(\mathcal{X})$, $T > 0, \eta > 0$, and $g_t, \mu_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta \left|g_t[x] - \mu_t[x]\right| \leq 1/4$. Then adaptive exponentiated gradient with losses g_t, predicted losses μ_t, and prior w satisfies:*

$$\sum_{t<T} g_t \cdot p_t \leq \sum_{t<T} g_t[x^*] + \eta \sum_{t<T} (g_t[x^*] - \mu_t[x^*])^2 + \eta^{-1} \log \frac{1}{w[x^*]}$$

We can combine this algorithm with the reduction in Section 2.2 in order to apply it to the case where experts sometimes decline to offer advice.

The only differences compared to Section 2.2 are to use Theorem 7 instead of Theorem 4 and and to translate the rewards such that the average reward is zero. That is, we still define

$$\widehat{p}_t[x] = \frac{p_t[x]z_t[x]}{\sum_{y\in\mathcal{X}} p_t[y]z_t[y]},$$

but now give the underlying adaptive OMD instance the losses

$$g_t[x] = z_t[x](\widehat{g}_t[x] - \overline{g_t})$$

and predicted losses

$$\mu_t[x] = z_t[x](\widehat{\mu}_t[x] - \overline{\mu_t}).$$

As before, we compute an approximation to $\overline{\mu_t}$ by using a fixed point argument and a binary search (in fact the value is exactly the same). We call this algorithm adaptive specialists.

Algorithm 3: $\textsc{Experts}_{\text{adapt}}$

```
function $\text{Init}_{\text{adapt}}(\mathcal{X}, w, \eta)$
    $g_{<0} \leftarrow 0$;
    $\beta_0 \leftarrow 0$;
    Return $S_0 = (\mathcal{X}, w, \eta, g_{<0}, \beta_0)$;
function $\text{Predict}_{\text{adapt}}(S_t = (\mathcal{X}, w, \eta, g_{<t}, \beta_t), z_t, \widehat{\mu}_t)$
    function $\widehat{p}(\overline{\mu})$
        $\mu[x] \leftarrow z_t[x](\widehat{\mu}_t[x] - \overline{\mu})$;
        $x \leftarrow \min_{x\in\Delta\mathcal{X}} (g_{<t} + \mu_t - \eta^{-1}\beta_t) \cdot x + \eta^{-1} D_{R_{\text{H}}}(x \parallel w)$;
        Return $\widehat{p}[x] = \frac{z[x]p[x]}{\sum_{y\in\mathcal{X}} z[y]p[y]}$;
    $\overline{\mu_t} \leftarrow$search for $\overline{\mu}$ such that $\overline{\mu} \approx \widehat{p}(\overline{\mu}) \cdot \widehat{\mu}_t$ ;
    Return $\widehat{p}_t = \widehat{p}(\overline{\mu_t})$;
function $\text{Update}_{\text{adapt}}(S_t = (\mathcal{X}, w, \eta, g_{<t}), z_t, \widehat{\mu}_t, \widehat{g}_t)$
    $\widehat{p}_t \leftarrow \text{Predict}_{\text{sleep}}(S_t, \widehat{\mu}_t, z_t)$;
    $\overline{g_t} \leftarrow \widehat{g}_t \cdot \widehat{p}_t$;
    $g_t[x] \leftarrow z_t[x](\widehat{g}_t[x] - \overline{g_t})$;
    $g_{<t+1} \leftarrow g_{<t} + g_t$;
    $\beta_{t+1}[x] \leftarrow \beta_t[x] + \log(1 - \eta g_t[x])$;
    Return $S_{t+1} = (\mathcal{X}, w, \eta, g_{<t+1}, \beta_{t+1})$;
```

Theorem 8. *For any $w \in \Delta(\mathcal{X})$, $T > 0, \eta > 0$, $z_t : \mathcal{X} \to [0,1]$, and $\widehat{g}_t, \widehat{\mu}_t : \mathcal{X} \to \mathbb{R}$ satisfying $\eta\,|\widehat{g}_t[x] - \widehat{\mu}_t[x]| \le 1/4$. Set $S_0 = \text{Init}_{\text{adapt}}(\mathcal{X}, w, \eta)$, $\widehat{p}_t = \text{Predict}_{\text{adapt}}(S_t, z_t, \widehat{\mu}_t)$,* *and* $S_{t+1} = \text{Update}_{\text{adapt}}(S_t, z_t, \widehat{\mu}_t, \widehat{g}_t)$*. For every* $x^* \in \mathcal{X}$*:*

$$\sum_{t<T} z_t[x^*]\widehat{g}_t \cdot \widehat{p}_t \leq \sum_{t<T} z_t[x^*]\widehat{g}_t[x^*] + \eta \sum_{t<T} z_t[x^*]^2(\widehat{g}_t[x^*] - \widehat{\mu}_t[x^*] + \overline{\mu_t} - \overline{g_t})^2 + \eta^{-1} \log \frac{1}{w[x^*]},$$

where $\overline{g_t} = \widehat{g}_t \cdot \widehat{p}_t$ *and* $\overline{\mu_t} = \widehat{\mu}_t \cdot \widehat{p}_t$*.*

Proof. Write $W_t = \sum_{x \in \mathcal{X}} z_t[x]p_t[x]$. We have:

$$\begin{aligned}
\sum_{t<T} g_t \cdot p_t &= \sum_{t<T} \sum_{x \in \mathcal{X}} z_t[x]p_t[x](\widehat{g}_t[x] - \overline{g_t}) \\
&= W \sum_{t<T} \sum_{x \in \mathcal{X}} \widehat{p}_t[x]\widehat{g}_t[x] - W\overline{g_t} \\
&= 0
\end{aligned}$$

Now we apply Theorem 7,

$$\begin{aligned}
\sum_{t<T} g_t \cdot p_t &\leq \sum_{t<T} g_t[x^*] + \eta \sum_{t<T} (g_t[x^*] - \mu_t[x^*])^2 + \eta^{-1} \log \frac{1}{w[x^*]} \\
-\sum_{t<T} g_t[x^*] &\leq \eta \sum_{t<T} (g_t[x^*] - \mu_t[x^*])^2 + \eta^{-1} \log \frac{1}{w[x^*]} \\
\sum_{t<T} z_t[x^*]\overline{g_t} - \sum_{t<T} z_t[x^*]\widehat{g}_t[x^*] &\leq \eta \sum_{t<T} z_t[x^*]^2(\widehat{g}_t[x^*] - \widehat{\mu}_t[x^*] - \overline{g_t} + \overline{\mu_t})^2 + \eta^{-1} \log \frac{1}{w[x^*]} \\
\sum_{t<T} z_t[x^*]\widehat{g}_t \cdot \widehat{p}_t &\leq \sum_{t<T} z_t[x^*]\widehat{g}_t[x^*] + \eta \sum_{t<T} z_t[x^*]^2(\widehat{g}_t[x^*] - \widehat{\mu}_t[x^*] - \overline{g_t} + \overline{\mu_t})^2 \\
&\quad + \eta^{-1} \log \frac{1}{w[x^*]},
\end{aligned}$$

as desired. □

Chapter 3

Online local learning

In preparation for our manipulation-resistant algorithms, we will first develop some machinery for online matrix learning based on semidefinite programming. These results were first described by the author in [10], and are a quantitative improvement over the previous matrix predictions results of [17].

In Section 3.1 we introduce the online local learning problem. In Section 3.2, we introduce the concept of pseudodistributions as a relaxation of probability distributions over discrete spaces, and in Section 3.3 we define the log determinant regularizer on this space and prove that it is strictly convex, which yields a statistically efficient algorithm for online local learning[1]. In Section 3.4 we present another application of the log determinant regularizer, showing that it allows us to solve a certain kind of multitask learning problem more efficiently by exploiting hints about what tasks are related to each other.

3.1 Model

In many learning problems we are interested in making predictions about the *relationships* amongst objects, which depend on latent properties of those objects. For example, we might want to predict which of two teams will win a game; or whether two people will get along; or whether a user's tastes are compatible with a particular movie.

We can model this as an online learning learning problem defined by a set $\mathcal{U}$ of objects and a set $\mathcal{V}$ of possible *labels* for those objects. For example, the objects might be teams and the labels may be measures of their skill, or the objects might be users and the labels might be "honest" or "dishonest." At each time step $t = 0, 1, \ldots$, the learner is given a pair of items $u_t^0, u_t^1 \in \mathcal{U}$ and must output a distribution p_t over pairs of labels $v_t^0, v_t^1 \in \mathcal{V}$. The learner then observes a loss function $\ell_t : \mathcal{V}^2 \to [-1, 1]$ and receives the loss $\ell_t(v_t^0, v_t^1)$. The learner's output may be stochastic, and all of our bounds will hold in expectation over the random choices of v_t^0, v_t^1.

[1] This result originally appeared in *Online Local Learning via Semidefinite Programming* at STOC 2014

The goal of the learner is to make predictions as good as the best fixed labeling $\mathbf{v} : \mathcal{U} \to \mathcal{V}$, i.e. to compete with quantities of the form $\sum_{t<T} \ell_t(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1))$.

Write $N = |\mathcal{U}|$ and $L = |\mathcal{V}|$.

In analogy with the model of optimistic learning discussed in Section 2.2, we can introduce a *predicted loss* $\mu_t : \mathcal{V}^2 \to [-1, 1]$. This gives our best guess for the loss function ℓ_t. Our goal is to obtain regret bounds that depend on the error $\sum_t \|\ell_t - \mu_t\|_\infty^2$ rather than on the squared losses $\ell_t(v^0, v^1)^2$ themselves.

We say that an algorithm has regret $\mathcal{R}(\mathbf{v})$ if, for every $T > 0, \eta > 0$, every sequence of loss functions $\ell_t : \mathcal{V}^2 \to [-1, 1]$ and every $\mathbf{v} : \mathcal{U} \to \mathcal{V}$, it outputs labels v_t^0, v_t^1 which satisfy (in expectation):

$$\sum_{t<T} \ell_t \cdot p_t \leq \sum_{t<T} \ell_t\big(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1)\big) + \eta \sum_{t<T} \max_{v^0, v^1} \big(\ell_t(v^0, v^1) - \mu_t(v^0, v^1)\big)^2 + \eta^{-1}\mathcal{R}(\mathbf{v}) \tag{3.1}$$

3.2 Pseudodistributions

We can view online local learning as a special case of online convex optimization, where the optimization is over the set of probability distributions over maps $\mathbf{v} : \mathcal{U} \to \mathcal{V}$. By using entropy regularization, we can obtain a regret of $\mathcal{R}(\mathbf{v}) = N \log L$.

The problem with this approach is that this space of probability distributions is very large. Indeed, even in hindsight it is NP hard to find the distribution which minimizes the total loss. Nevertheless, we might hope to find some efficient algorithm that makes predictions that are competitive with those of the best fixed $\mathbf{v}$ (though such an algorithm will necessarily be making predictions that aren't consistent with any particular $\mathbf{v}$).

Our first observation is that we don't need to actually keep track of an entire distribution over maps $\mathbf{v} : \mathcal{U} \to \mathcal{V}$. All we need to know is the marginal distribution of the pairs $(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1))$. We can record these marginal distributions in a matrix X with rows and columns indexed by $\mathcal{U} \times \mathcal{V}$, where the $X_{(u_t^0, v_t^0),(u_t^1, v_t^1)}$ entry is the probability that $\mathbf{v}(u_t^0) = v_t^0$ and $\mathbf{v}(u_t^1) = v_t^1$.

Matrices that arise as actual marginals of a distribution clearly satisfy a number of "local" properties:

- For every $u^0, u^1 \in \mathcal{U}$: $\sum_{v^0, v^1 \in \mathcal{V}} X_{(u^0, v^0),(u^1, v^1)} = 1$.
- For every $u^0, u^1 \in \mathcal{U}, v^0, v^1 \in \mathcal{V}$: $X_{(u^0, v^0)(u^1, v^1)} = X_{(u^1, v^1)(u^0, v^0)}$.
- For every $u \in \mathcal{U}, v^0, v^1 \in \mathcal{V}$: $X_{(u, v^0),(u, v^1)}$ is 1 if $v^0 = v^1$, and 0 otherwise.
- For every $u^0, u^1 \in \mathcal{U}, v^0 \in \mathcal{V}$: $\sum_{v^1 \in \mathcal{V}} X_{(u^0, v^0),(u^1, v^1)} = X_{(u^0, v^0),(u^0, v^0)}$.
- Every entry of X is nonnegative.

These constraints defines the first level of the Sherali-Adams hierarchy [27]; we will write $\mathcal{M}$ for the set of matrices that satisfy these properties, and call them "pseudomarginals."

Given some pseudomarginals X and a pair of objects u_t^0, u_t^1, we can define $p_t(v^0, v^1) = X_{(u_t^0, v_t^0),(u_t^1, v_t^1)}$. The properties above guarantee that this is a well-defined distribution over $\mathcal{V}^2$. The loss of a pseudomarginal, $\ell_t(X)$, is then $\sum_{v^0,v^1} \ell_t(v^0, v^1) X_{(u_t^0,v^0),(u_t^1,v^1)}$.

If we could do online convex optimization over the set $\mathcal{M}$, this could be even better than doing online convex optimization over the actual set of marginals of distributions over maps $\mathbf{v} : \mathcal{U} \to \mathcal{V}$. The problem with this idea is that the set $\mathcal{M}$ of pseudomarginals is actually *much* larger than the set of all actual marginals, and as a result it is impossible to get low regret in the corresponding online convex optimization problem: it is easy to prove that the best possible regret over T rounds is $\Omega\left(\sqrt{TN^2}\right)$.

To improve the situation, we need to find a smaller set of matrices. We want the set to be small so that it is possible to do online convex optimization with low regret. At the same time we want to ensure that the set contains all of the actual marginals, so that competing with the best pseudomarginal is at least as good as competing the best marginal.

Fortunately there is another easy-to-verify property of a real marginal distribution: the matrix X should be positive semi-definite. So rather than considering the set $\mathcal{M}$, we can consider the subset $\mathcal{M}^+ \subset \mathcal{M}$ that are also positive semi-definite. This corresponds to the Lasserre hierarchy rather than the Sherali-Adams hierarchy [20]. It will turn out that this set is small enough to be useful for our purposes.

3.3 The log determinant regularizer

We will apply online mirror descent on the set $\mathcal{M}^+$, using an appropriate convex regularizer $R : \mathcal{M}^+ \to \mathbb{R}$. We will then apply Theorem 1, which requires R to satisfy an appropriate strong convexity property.

Entropy regularization is often effective for regularizing spaces of probability distributions, since the entropy is strongly convex yet its value grows logarithmically with the "size" of the space. If we could optimize over the space of matrices that actually arise as the marginals of a distribution over $\mathbf{v} : \mathcal{U} \to \mathcal{V}$, then we could use as our regularizer the maximum entropy of any distribution consistent with the given marginals. It's not clear what the appropriate analog is in the case of pseudomarginals that belong to $\mathcal{M}^+$, since such pseudomarginals need not correspond to any actual distribution.

One of the most powerful properties of pseudomarginals in $\mathcal{M}^+$ is that they *do* arise as the moments of a family of *real-valued* random variables, one for each pair $(u, v) \in \mathcal{U} \times \mathcal{V}$, even if they don't correspond to any distribution over *0-1-valued* random variables. Sampling from this distribution doesn't actually produce a map $\mathcal{U} \to \mathcal{V}$, because some of the "probabilities" are negative numbers. Nevertheless, this gives us a tool to relate the local data in $\mathcal{M}^+$ to some kind of global structure. This connection has proved to be extremely useful in the context of approximation algorithms for constraint satisfaction problems [23].

This connection also gives us a natural analog of the entropy regularization for pseudomarginals. Namely, for any pseudomarginal $X \in \mathcal{M}^+$, we can consider the maximum (differential) entropy of any continuous distribution which is consistent with a given pseudomarginal. Conveniently, the maximum entropy distribution is a Gaussian, whose entropy can be easily computed as the log determinant of the matrix of moments. In fact, exactly the same argument that shows that entropy regularization is strongly convex will show that this regularization is strongly convex.

Define

$$R_{\mathrm{logdet}}(X) = -\log\det(LX + I)$$

Lemma 6. *For all $X \in \mathcal{M}^+$,*

$$-LN \log 2 \le R_{\mathrm{logdet}}(X) \le 0.$$

Proof. Note that $\log\det(LX + I) = \sum \log(L\lambda_i + 1)$, where λ_i are the eigenvalues of X. Since $X \succeq 0$, all $\lambda_i \ge 0$, and hence $\log\det(LX + I) \ge 0$.

On the other hand, $\sum_i \lambda_i = \operatorname{Tr} X = N$. Moreover, $\log(L\lambda + 1)$ is a convex function of λ, so by Jensen's inequality the sum is maximized when $\lambda_1 = \ldots = \lambda_{LN} = \frac{1}{L}$. Thus $\sum \log(L\lambda_i + 1) \le LN \log\bigl(L\frac{1}{L} + 1\bigr) = LN \log 2$. □

To analyze the convexity of this regularizer, we introduce two new norms:

1. $\|X\|_{\infty,1} = \max_{u^0,u^1} \sum_{v^0,v^1} \left|X_{(u^0,v^0)(u^1,v^1)}\right|$.
2. $\|X\|_{1,\infty} = \sum_{u^0,u^1} \max_{v^0,v^1} \left|X_{(u^0,v^0)(u^1,v^1)}\right|$.

It is trivial to verify to that $\|\cdot\|_{1,\infty}$ and $\|\cdot\|_{\infty,1}$ are dual norms.

In [10], I proved a bound on the convexity of R_{logdet} by relating it to the entropy of Gaussians. In [5], the authors prove a stronger bound by a direct calculation of the inverse Hessian. (They bound $g_t^T(\nabla^2 R_{\mathrm{logdet}})^{-1} g_t$ in terms of $\|g_t\|_{1,\infty}$. This is easily seen to imply a similar bound on the strong convexity of R_{logdet}.)

Lemma 7 (Section 3.2 of [5])**.** *R_{logdet} is 1-strongly-convex in the norm $\|\cdot\|_{\infty,1}$.*

Combining this with Theorem 1, we obtain:

Theorem 9. *Fix finite sets $\mathcal{U}$ and $\mathcal{V}$ of sizes N and L respectively, and any $T > 0, \eta > 0$, $\mathbf{v} : \mathcal{U} \to \mathcal{V}$, $u_t^0, u_t^1 \in \mathcal{U}$, $\ell_t, \mu_t : \mathcal{V}^2 \to [-1, 1]$. Set $S_0 = \mathrm{Init}_{\mathrm{local}}(\mathcal{U}, \mathcal{V}, R_{\mathrm{logdet}})$, $p_t = \mathrm{Predict}_{\mathrm{local}}(S_t, u_t^0, u_t^1, \mu_t)$, and $S_{t+1} = \mathrm{Update}_{\mathrm{local}}(S_t, u_t^0, u_t^1, \mu_t, \ell_t)$. Then for every $\mathbf{v} : \mathcal{U} \to \mathcal{V}$:*

$$\sum_{t<T} \ell_t \cdot p_t \le \sum_{t<T} \ell_t\bigl(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1)\bigr) + \eta \sum_{t<T} \|\ell_t - \mu_t\|_\infty^2 + \eta^{-1} LN \log 2.$$

Proof. In expectation, $\ell_t(v_t^0, v_t^1)$ is precisely equal to $g_t \cdot X_t$. Moreover, X_t is produced by OMD with the losses g_t and predicted losses M_t. We can verify by inspection that

Algorithm 4: $\textsc{LocalLearning}(R)$

```
function $\mathrm{Init}_{\mathrm{local}}(\mathcal{U}, \mathcal{V}, R)$
    $g_{<0} \leftarrow 0 \in \mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$;
    Return $S_0 = (R, g_{<0})$;
function $\mathrm{Predict}_{\mathrm{local}}(S_t = (R, g_{<t}), u_t^0, u_t^1, \mu_t)$
    $M_t \leftarrow 0 \in \mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$;
    for $v^0, v^1 \in \mathcal{V}$ do
        $(M_t)_{(u_t^0,v_t^0)(u_t^1,v_t^1)} \leftarrow \mu_t(v_t^0, v_t^1)$;
    $X_t \leftarrow \arg\min_{X \in \mathcal{M}^+}((g_{<t} + M_t) \cdot X + R(X))$;
    Return $p_t = (v_t^0, v_t^1) \to (X_t)_{(u_t^0,v_t^0)(u_t^1,v_t^1)} \in \Delta(\mathcal{V}^2)$;
function $\mathrm{Update}_{\mathrm{local}}(S_t = (R, g_{<t}), u_t^0, u_t^1, \mu_t, \ell_t)$
    $g_t \leftarrow 0 \in \mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$;
    for $v^0, v^1 \in \mathcal{V}$ do
        $(g_t)_{(u_t^0,v_t^0)(u_t^1,v_t^1)} \leftarrow \ell_t(v_t^0, v_t^1)$;
    $g_{<t+1} \leftarrow g_{<t} + g_t$;
    Return $S_{t+1} = (R, g_{<t+1})$;
```

$\|g_t - M_t\|_{1,\infty} = \|\ell_t - \mu_t\|_\infty$. By Theorem 1 we have for any $X^* \in \mathcal{M}^+$:

$$\begin{aligned}\sum_{t<T} g_t \cdot X_t &\le g_{<T} \cdot X^* + \eta \sum_{t<T} \|g_t - M_t\|_{1,\infty}^2 + \eta^{-1} LN \log 2 \\ &= g_{<T} \cdot X^* + \eta \sum_{t<T} \|\ell_t - \mu_t\|_\infty^2 + \eta^{-1} LN \log 2.\end{aligned}$$

In particular, we can take X^* to the matrix with $X^*_{(u^0,v^0)(u^1,v^1)} = 1$ if $\mathbf{v}(u^0) = v^0$ and $\mathbf{v}(u^1) = v^1$, and 0 otherwise. Then $g_{<T} \cdot X^* = \sum_{t<T} \ell_t(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1))$, and we obtain the desired result. □

[5] show that qualitatively improving this bound would require identifying planted cliques of size $o(N^{1/2})$. Identifying such cliques is beyond the reach of current techniques, and some evidence suggests it may be computationally intractable [12].

3.4 Local learning with relationships

We are ultimately interested in collaborative learning problems, where the objects $u_t \in \mathcal{U}$ are users. In these learning problems, we may have side information about which users are related to each other, and we might suspect that related users are more likely to have the same label. For example, we may know which pairs of users are friends on a social network, and expect honest users to mostly be friends with other honest users.

We can represent this information as a graph E, where two users u^0 and u^1 who are adjacent in E are more likely to have the same label under a good labeling $\mathbf{v}$. This information can be exploited by extending the techniques laid out in the previous sections. We will assume that E is k-regular; this isn't important for our algorithms, but it makes it much easier to state our bounds.

Algorithm 4 achieves regret $\mathcal{R}(\mathbf{v}) = LN$. We will show how to improve this bound to $\mathcal{R}(\mathbf{v}) = \mathcal{O}\big(LN\big(\alpha + \frac{\log k}{k}\big)\big)$, where α is the fraction of edges $(u^0, u^1) \in E$ such that $\mathbf{v}(u^0) \neq \mathbf{v}(u^1)$. That is, if most pairs of connected users have the same value under $\mathbf{v}$, then we are able to significantly improve our regret. If the connections are taken at random, then we recover the previous result $\mathcal{R}(\mathbf{v}) = \mathcal{O}(LN)$. But if most connected users have the same image under $\mathbf{v}$ then this bound can be much tighter.

This modified regret bound is also achieved by OMD over $\mathcal{M}^+$, but with a different regularizer. Define A_E as the adjacency matrix of the graph on $\mathcal{U} \times \mathcal{V}$ with an edge between (u^0, v^0) and (u^1, v^1) iff $(u^0, u^1) \in E$ and $v^0 = v^1$. We will use the regularizer:

$$R_E(X) = -\frac{L}{k}\operatorname{Tr}(A_E X) - \log\det(LX + I).$$

R_E differs by a linear function from the log det regularizer R_{logdet}, and so it satisfies the same strong convexity bound Lemma 7. The only change in the analysis are the bounds established in Lemma 6. Before bounding the values of this regularizer, we prove a useful algebraic identity:

Lemma 8. *For any $k \geq e, \lambda \geq 0, \mu \leq 1$:*

$$\lambda\mu + \log(1 + \lambda) \leq 16\mu^2 \log k + \left(1 + \frac{1}{k}\right)\lambda$$

Proof. Case 1: If $\lambda \leq 1$, then $\log(1 + \lambda) \leq \lambda - \lambda^2/4$. By the arithmetic mean geometric mean inequality, $\lambda\mu \leq \frac{1}{2}(\lambda^2/2 + 2\mu^2)$. Thus:

$$\begin{aligned} \lambda\mu + \log(1 + \lambda) &\leq \lambda\mu + \lambda - \lambda^2/4 \\ &\leq \mu^2 + \lambda \end{aligned}$$

Case 2: If $\mu > 1/4$, then we have

$$\begin{aligned} \mu\lambda + \log(1 + \lambda) &= \mu\lambda + \log k + \log\left(\frac{1 + \lambda}{k}\right) \\ &< \mu\lambda + \log k + \log(1 + \lambda k) \\ &< \mu\lambda + \log k + \frac{\lambda}{k} \\ &\leq \lambda + \log k + \frac{\lambda}{k} \\ &\leq 16\mu^2 \log k + \left(1 + \frac{1}{k}\right)\lambda \end{aligned}$$

Case 3: If $\mu \leq 1/4$ and $\lambda > 1$, then $\lambda\mu \leq \lambda/4$ and $\log(1+\lambda) \leq \log(2)\lambda$. Thus

$$\lambda\mu + \log(1+\lambda) \leq (0.25 \log 2)\lambda < \lambda$$

□

We now prove:

Lemma 9. *For any $X \in \mathcal{M}^+$ we have*

$$-\left(1 + \mathcal{O}\left(\frac{\log k}{k}\right)\right)LN \leq R_E(X) \leq -\frac{L}{k}\operatorname{Tr}(A_E X).$$

Proof. The upper bound follows immediately from Lemma 6.

For the lower bound, let λ_i be the eigenvalues of X with corresponding orthonormal eigenvectors w_i, such that $X = \sum_i \lambda_i w_i w_i^T$. For any vector w:

$$\begin{aligned} w^T A_E w &= \sum_{\substack{(u,u')\in E \\ v\in\mathcal{V}}} w_{u,v} w_{u',v} \\ &\leq \frac{1}{2} \sum_{\substack{(u,u')\in E \\ v\in\mathcal{V}}} \left(w_{u,v}^2 + w_{u',v}^2\right) \\ &= k \sum_{\substack{u\in\mathcal{U} \\ v\in\mathcal{V}}} w_{u,v}^2 \end{aligned}$$

Thus $\frac{w_i^T A_E w_i}{k} \leq 1$, so we can apply Lemma 8:

$$\begin{aligned} R_E(X) &= -\sum_i \left(\log(1 + L\lambda_i) + \frac{1}{k} L\lambda_i w_i^T A_E w_i\right) \\ &\leq -16 \log k \sum_i \left(\frac{w_i^T A_E w_i}{k}\right)^2 - L\left(1 + \frac{1}{k}\right)\sum_i \lambda_i \end{aligned}$$

But $\sum \lambda_i = \operatorname{Tr} X \leq N$, and $\sum \left(v_i^T A v_i\right)^2 \leq \operatorname{Tr} A_E^2 = LNk$. Thus

$$R_E(X) \geq -16 \log k \frac{LN}{k} - \left(1 + \frac{1}{k}\right)LN = -N - \mathcal{O}\left(\frac{N \log k}{k}\right)$$

as desired.

□

R_E is 1-strongly-convex in the $\|\cdot\|_{\infty,1}$ because it differs by a linear term from R_{logdet}. By applying Theorem 1, we obtain:

Theorem 10. *Fix $\mathcal{U}$ and $\mathcal{V}$, let E a k-regular graph on $\mathcal{U}$ $X^* \in \mathcal{M}^+$, and define*

$$\begin{aligned}\alpha &= 1 - \frac{1}{Nk}\operatorname{Tr}(A_E X^*)\\ &= 1 - \mathbb{E}_{(u^0,u^1)\in E}\left[\sum_{v\in\mathcal{V}} X^*_{(u^0,v)(u^1,v)}\right]\end{aligned}$$

Fix any $T > 0, \eta > 0$, $g_t, M_t \in \mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$. optimistic OMD over $\mathcal{M}^+$ with regularizer R_E, losses g_t, and predicted losses M_t satisfies:

$$\sum_{t<T} g_t \cdot X_t \le g_{<T} \cdot X^* + \eta \sum_{t<T} \|g_t - M_t\|_{1,\infty}^2 + \mathcal{O}\left(\eta^{-1} NL\left(\alpha + \frac{\log k}{k}\right)\right)$$

And by applying this to online local learning, we obtain:

Theorem 11. *Fix $\mathcal{U}$ and $\mathcal{V}$, and let E be a k-regular graph on $\mathcal{U}$. Fix any $T > 0, \eta > 0$, $u_t^0, u_t^1 \in \mathcal{U}$, $\ell_t, \mu_t : \mathcal{V}^2 \to [-1, 1]$. Set $S_0 = \mathrm{Init}_{\mathrm{local}}(\mathcal{U}, \mathcal{V}, R_E)$, $p_t = \mathrm{Predict}_{\mathrm{local}}(S_t, u_t^0, u_t^1, \mu_t)$, and $S_{t+1} = \mathrm{Update}_{\mathrm{local}}(S_t, u_t^0, u_t^1, \mu_t, \ell_t)$. Then for every $\mathbf{v} : \mathcal{U} \to \mathcal{V}$:*

$$\sum_{t<T} \ell_t(v_t^0, v_t^1) \le \sum_{t<T} \ell_t(\mathbf{v}(u_t^0), \mathbf{v}(u_t^1)) + \eta \sum_{t<T} \|\ell_t - \mu_t\|_\infty^2 + \mathcal{O}\left(\eta^{-1} NL\left(\alpha_E(\mathbf{v}) + \frac{\log k}{k}\right)\right)$$

where $\alpha_E(\mathbf{v}) = \mathbb{P}_{(u^0,u^1)\in E}(\mathbf{v}(u^0) \ne \mathbf{v}(u^1))$.

Proof. The proof exactly follows the proof of Theorem 3.1. The only observation is that if we take X^* to be the indicator matrix for $\mathbf{v}$, then $\alpha_E(\mathbf{v}) = 1 - \frac{1}{Nk}\operatorname{Tr}(A_E X^*)$, so that we can apply Theorem 10 to obtain the desired result. □

We are not aware of any statistical obstruction to obtaining regret $LN\left(\alpha + \frac{1}{k}\right)$. It is easy to see that the LN/k term in the regret is necessary—if E consists of N/k clusters of size k, and if only one user from each cluster is ever involved in an interaction, then we are back in the general local learning setup.

Chapter 4

Bandits and experts

In this chapter we will study collaborative versions of two foundational problems in online learning: prediction with expert advice and contextual bandits.

4.1 Model

Both problems in this chapter involve a finite set $\mathcal{U}$ of users and a finite set of experts $\mathcal{X}$.

Prediction with expert advice

In each round $t = 0, 1, 2, \ldots$ of collaborative prediction with expert advice, we are given a user $u_t \in \mathcal{U}$ and must select an expert $x_t \in \mathcal{X}$ (perhaps stochastically). The learner then observes the loss function $\ell_t : \mathcal{X} \to [-1, 1]$. We define the loss

$$\ell_{<T}(H) = \sum_{\substack{t<T \\ u_t \in H}} \ell_t(x_t)$$

and the benchmark

$$\mathrm{OPT}_{<T}(H) = \min_{x^* \in \mathcal{X}} \sum_{\substack{t<T \\ u_t \in H}} \ell_t(x^*).$$

Our goal is to minimize the regret $\ell_{<T}(H) - \mathrm{OPT}_{<T}(H)$.

Contextual bandits

In the collaborative contextual bandits problem we introduce a set of "arms" or actions $\mathcal{A}$. In round t we are given the identity of a user $u_t \in \mathcal{U}$, and for each expert $x \in \mathcal{X}$ we are given a distribution $q_t^x \in \Delta(\mathcal{A})$. Nature selects a loss function $\ell_t : \mathcal{A} \to [0, 1]$, but does not reveal it. We then pick an action a_t stochastically, and receive and observe the loss $\ell_t(a_t)$.

Our benchmark is now the loss that would be obtained by following the recommendation of the best expert x^*:

$$\mathrm{OPT}_{<T}(H) = \min_{x^* \in \mathcal{X}} \sum_{\substack{t<T \\ u_t \in H}} \mathbb{E}_{a \sim q_t^{x^*}} [\ell_t(a)]$$

4.2 Prediction with expert advice

Our algorithm for collaborative prediction with expert advice follows the plan laid out in Section 1.2. We first define a "filtering" problem faced by an expert choosing when to offer advice; then show how to use an algorithm for the filtering problem to solve collaborative prediction with advice; and finally in Section 4.3 describe three algorithms for the filtering problem.

The filtering problem

In our filtering problem, there is a set of users $\mathcal{U}$ and a sequence of rounds $t = 0, 1, \ldots$. In each round a single user $u_t \in \mathcal{U}$ is active, and the learner must output $z_t \in [0, 1]$. The learner then observes a loss $\ell_t \in \mathbb{R}$.

The learner's goal is to minimize the loss

$$\ell_{<T} = \sum_{t<T} z_t \ell_t.$$

The benchmark strategies are those that pick a fixed set of users $H \subset \mathcal{U}$ and then output $z_t = 1$ precisely when $u_t \in H$:

$$\ell_{<T}(H) = \sum_{\substack{t<T \\ u_t \in H}} \ell_t.$$

The learner's goal is to minimize the difference between their loss and the best loss of the form $\ell_{<T}(H)$. The *regret* of a learning algorithm on a set H is defined as the maximum, over all sequences of loss functions, of $\ell_{<T} - \ell_{<T}(H)$. As in Chapter 2, we assume that before choosing z_t the learner has access to some prediction μ_t about the value of the loss ℓ_t (we can take $\mu_t = 0$ if the learner has no information about ℓ_t).

We say that an algorithm is weakly competitive with regret $\mathcal{R}(H)$ if it satisfies:

$$\ell_{<T} \leq \ell_{<T}(H) + \eta \sum_{t<T} (\ell_t - \mu_t)^2 + \frac{\mathcal{R}(H)}{\eta}, \tag{4.1}$$

where η is the learning ate. We say that it is strongly competitive if:

$$\ell_{<T} \leq \ell_{<T}(H) + \eta \sum_{t<T} z_t (\ell_t - \mu_t)^2 + \frac{\mathcal{R}(H)}{\eta}. \tag{4.2}$$

Our goal is to find competitive algorithms where $\mathcal{R}(H)$ is as small as possible.

An algorithm for this problem is defined by three methods:

- $S_0 = \text{Init}_{\text{filter}}(\mathcal{U}, \eta)$ returns the initial state of the algorithm.
- $z_t = \text{Predict}_{\text{filter}}(S_t, u_t, \mu_t)$ outputs z_t, given u_t and the current state S_t of the algorithm.
- $S_{t+1} = \text{Update}_{\text{filter}}(S_t, u_t, \mu_t, \ell_t)$ outputs the next state of the algorithm.

Our algorithm for prediction with expert advice

Given an algorithm Filter for the filtering problem defined in the previous section, we can obtain an algorithm Collab(Filter) for collaborative prediction with expert advice. In this section we will leave Filter unspecified; in the next section we introduce three algorithms. To get a quantitative feel for the results, note that the regret $\mathcal{R}(H)$ in Equations 4.1 and 4.2 are $\mathcal{O}(|\mathcal{U}|)$.

Algorithm 5: Collab(Filter)

```
$S_0 \leftarrow \text{Init}_{\text{adapt}}(\mathcal{X}, \text{uniform}, \eta)$;
for $x \in \mathcal{X}$ do
    $S_0^x \leftarrow \text{Init}_{\text{filter}}(\mathcal{U}, \eta)$;
for $t = 0, 1, 2, \ldots$ do
    Observe $u_t \in \mathcal{U}$;
    for $x \in \mathcal{X}$ do
        $z_t(x) \leftarrow \text{Predict}_{\text{filter}}(S_t^x, u_t, 8\eta)$;
    Play $p_t(x) = \text{Predict}_{\text{adapt}}(S_t, z_t, \mathbf{0})$;
    Observe $\ell_t : \mathcal{X} \to [-1, 1]$;
    for $x \in \mathcal{X}$ do
        $\ell_t^x \leftarrow \ell_t(x) - \ell_t \cdot p_t + 8\eta$;
        $S_{t+1}^x \leftarrow \text{Update}_{\text{filter}}(S_t^x, u_t, 8\eta, \ell_t^x)$;
    $S_{t+1} \leftarrow \text{Update}_{\text{adapt}}(S_t, z_t, \mathbf{0}, \ell_t)$;
```

Theorem 12. *Let $\mathcal{X}$ and $\mathcal{U}$ be arbitrary. Suppose that* Filter *is weakly competitive with regret $\mathcal{R}(H)$, i.e. satisfies Equation 4.1 (respectively, that* Filter *is strongly competitive, i.e. satisfies Equation 4.2). Then for any $H \subset \mathcal{U}$, $x \in \mathcal{X}$, $T > 0, \eta > 0, \ell_t : \mathcal{X} \to [-1, 1]$,* Collab(Filter) *satisfies*

$$\sum_{t \leq T : u_t \in H} \ell_t \cdot p_t \leq \sum_{t \leq T : u_t \in H} \ell_t(x) + \mathcal{O}\left(\eta T^\star + \frac{\log |\mathcal{X}| + \mathcal{R}(H)}{\eta}\right)$$

where $T^\star = T$ (respectively $T^\star = \# \{t < T : u_t \in H\}$).

Proof. Let $z_t^\star(x) = 1$ if Filter is weakly competitive, and $z_t^\star(x) = z_t(x)$ if Filter is strongly competitive. By checking cases, we can easily verify the two inequalities:

$$\begin{aligned}\sum_{t<T} z_t^\star(x) &\le \sum_{t<T} z_t(x) + T^\star \\ \sum_{\substack{t<T\\u_t\in H}} \eta &\le \eta T^\star.\end{aligned}$$

Write $r_t(x) = \ell_t(x) - \ell_t \cdot p_t$.

By applying Equation 4.1 (respectively Equation 4.2) to S_t^x, we obtain:

$$\begin{aligned}\sum_{t<T} z_t(x)\ell_t^x &\le \sum_{\substack{t<T\\u_t\in H}} \ell_t^x + \eta \sum_{t<T} z_t^\star(x)(\ell_t^x - 8\eta)^2 + \frac{\mathcal{R}(H)}{\eta} \\ \sum_{t<T} z_t(x)(r_t(x) + 8\eta) &\le \sum_{\substack{t<T\\u_t\in H}} (r_t(x) + 8\eta) + \eta \sum_{t<T} z_t^\star(x) r_t(x)^2 + \frac{\mathcal{R}(H)}{\eta} \\ &\le \sum_{\substack{t<T\\u_t\in H}} r_t(x) + 8\eta T^\star + 4\eta \sum_{t<T} z_t^\star(x) + \frac{\mathcal{R}(H)}{\eta} \\ &\le \sum_{\substack{t<T\\u_t\in H}} r_t(x) + 8\eta T^\star + 4\eta \sum_{t<T} z_t(x) + 4\eta T^\star + \frac{\mathcal{R}(H)}{\eta} \\ \sum_{t<T} z_t(x)(r_t(x) + 4\eta) &\le \sum_{\substack{t<T\\u_t\in H}} r_t(x) + \mathcal{O}(\eta T^\star) + \frac{\mathcal{R}(H)}{\eta}.\end{aligned}$$

By applying Theorem 8 to S_t with expert x:

$$\begin{aligned}-\sum_{t<T} z_t(x) r_t(x) &\le \eta \sum_{t<T} z_t(x) r_t(x)^2 + \frac{\log|\mathcal{X}|}{\eta} \\ &\le 4\eta \sum_{t<T} z_t(x) + \frac{\log|\mathcal{X}|}{\eta} \\ -z_t(x)(r_t(x) + 4\eta) &\le \frac{\log|\mathcal{X}|}{\eta}\end{aligned}$$

Combining these two inequalities, we obtain:

$$\begin{aligned}\sum_{\substack{t<T\\u_t\in H}} \ell_t \cdot p_t &= \sum_{\substack{t<T\\u_t\in H}} \ell_t(x) - \sum_{\substack{t<T\\u_t\in H}} r_t(x)\\ &\leq \sum_{\substack{t<T\\u_t\in H}} \ell_t(x) + \mathcal{O}(\eta T^\star) + \frac{\mathcal{R}(H)}{\eta} - \sum_{t<T} z_t(x)(r_t(x)+4\eta)\\ &\leq \sum_{\substack{t<T\\u_t\in H}} \ell_t(x) + \mathcal{O}(\eta T^\star) + \frac{\mathcal{R}(H)}{\eta} + \frac{\log|\mathcal{X}|}{\eta},\end{aligned}$$

as desired. □

4.3 Algorithms for the filtering problem

Treating users separately

The simplest algorithm for learning H is to consider each expert separately, and to pick some prior probability θ that each expert is honest. Algorithm 6 implements this strategy.

Algorithm 6: FILTER$_\theta$

```
function Init_θ($\mathcal{U}, \eta$)
    $w(1) \leftarrow \theta$;
    $w(0) \leftarrow 1 - \theta$;
    for $u \in \mathcal{U}$ do
        $S_0^u \leftarrow \mathrm{Init}_{\mathrm{sleep}}(\{0,1\}, w, \eta)$;
    Return $S_0 = (S_0^u)_{u\in\mathcal{U}}$;
function Predict_filter$\big(S_t = (S_t^u)_{u\in\mathcal{U}}, u_t, \mu_t\big)$
    Return $z_t = \mathrm{Predict}_{\mathrm{sleep}}(S_t^{u_t}, \mathbf{1}, z \mapsto z\mu_t)[1]$;
function Update_filter$\big(S_t = (S_t^u)_{u\in\mathcal{U}}, u_t, \ell_t, \mu_t\big)$
    $S_{t+1}^{u_t} \leftarrow \mathrm{Update}_{\mathrm{sleep}}(S_t^u, \mathbf{1}, z \mapsto z\mu_t, z \mapsto z\ell_t)$;
    for $u \in \mathcal{U}\setminus\{u_t\}$ do
        $S_{t+1}^u \leftarrow S_t^u$;
    Return $S_{t+1} = \big(S_{t+1}^u\big)_{u\in\mathcal{U}}$;
```

Lemma 10. *Fix a finite set $\mathcal{U}$ and $\theta \in [0,1]$. Fix any $H \subset \mathcal{U}$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in \mathbb{R}$, and define $S_0 = \mathrm{Init}_\theta(\mathcal{U}, \eta)$, $z_t = \mathrm{Predict}_\theta(S_t, u_t, \mu_t)$, $S_{t+1} = \mathrm{Update}_\theta(S_t, u_t, \mu_t, \ell_t)$. Then we* *have:*

$$\sum_{t<T} z_t \ell_t \leq \sum_{\substack{t<T \\ u_t \in H}} \ell_t + \eta \sum_{t<T} z_t (\ell_t - \mu_t)^2 + \eta^{-1} \log \frac{1}{p_\theta(H)},$$

where $p_\theta(H) = \theta^{|H|}(1-\theta)^{|\mathcal{U} \setminus H|}$.

That is, Algorithm 6 is strongly competitive (satisfies Equation 4.2) with $\mathcal{R}(H) = -\log p_\theta(H)$.

Proof. Let $\chi_H(u)$ be 1 if $u \in H$ and 0 otherwise. By applying Lemma 5 to S_t^u, we obtain:

$$\sum_{\substack{t<T \\ u_t = u}} z_t \ell_t \leq \sum_{\substack{t<T \\ u_t = u}} \chi_H(u) \ell_t + \eta \sum_{\substack{t<T \\ u_t = u}} z_t (\ell_t - \mu_t)^2 + \eta^{-1}(\chi_H(u) \log \theta + (1 - \chi_H(u)) \log(1-\theta)).$$

Summing these inequalities up across all u, we obtain the desired result. □

Theorem 13. *For any finite sets* $\mathcal{U}$ *and* $\mathcal{X}$, $H \subset \mathcal{U}$, $x^* \in \mathcal{X}$, $\theta \in [0,1]$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in [-1,1]$, Collab(Filter$_\theta$) *with losses* ℓ_t *and predicted losses* μ_t *satisfies:*

$$\sum_{\substack{t<T \\ u_t \in H}} \ell_t \cdot p_t \leq \sum_{\substack{t<T \\ u_t \in H}} \ell_t(x^*) + \eta T_H + \mathcal{O}\left(\eta^{-1}\left(\log \frac{1}{p_\theta(H)} + \log |\mathcal{X}|\right)\right)$$

where $T_H = |\{t < T : u_t \in H\}|$.

Treating users separately and competing weakly

The algorithm in the previous section was *strongly competitive*, i.e. satisfies Equation 4.2. When the set H is reasonably large, we can obtain a smaller value of $\mathcal{R}(H)$ if we are satisfied with being weakly competitive, merely satisfying Equation 4.1.

We can do this by treating each user as a separate convex optimization problem over the space $\mathcal{X} = [0,1]$, and using a regularizer which is minimized at 1.

Lemma 11. *Fix a finite set* $\mathcal{U}$, $H \subset \mathcal{U}$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in \mathbb{R}$, *and define* $S_0 = \mathrm{Init}_{\mathrm{weak}}(\mathcal{U}, \eta)$, $z_t = \mathrm{Predict}_{\mathrm{weak}}(S_t, u_t, \mu_t)$, $S_{t+1} = \mathrm{Update}_{\mathrm{weak}}(S_t, u_t, \mu_t, \ell_t)$. *Then we have:*

$$\sum_{t<T} z_t \ell_t \leq \sum_{\substack{t<T \\ u_t \in H}} \ell_t + \eta \sum_{t<T} z_t (\ell_t - \mu_t)^2 + \eta^{-1} |\mathcal{U} \setminus H|.$$

That is, Algorithm 6 is weakly competitive (satisfies Equation 4.1) with $\mathcal{R}(H) = |\mathcal{U} \setminus H|$.

Proof. Let $\chi_H(u)$ be 1 if $u \in H$ and 0 otherwise.

Note that $z \mapsto (1-z)^2$ is 1-strongly convex. So by Lemma 2 its Frenchel conjugate is 1-strongly smooth, and we can apply Theorem 1 to each user u and obtain:

$$\sum_{\substack{t<T \\ u_t = u}} z_t \ell_t \leq \sum_{\substack{t<T \\ u_t = u}} \chi_H(u) \ell_t + \eta \sum_{\substack{t<T \\ u_t = u}} (\ell_t - \mu_t)^2 + \eta^{-1}(1 - \chi_H(u))^2$$

Summing these inequalities up across all u, we obtain the desired result. □

Algorithm 7: $\textsc{Filter}_{\text{weak}}$

```
function $\text{Init}_{\text{weak}}(\mathcal{U}, \eta)$
    for $u \in \mathcal{U}$ do
        $g^u_{<0} \leftarrow 0$;
    Return $S_0 = (g^u_{<0})_{u \in \mathcal{U}}$;
function $\text{Predict}_{\text{weak}}\big(S_t = (g^u_{<t})_{u \in \mathcal{U}}, u_t, \mu_t\big)$
    Return $z_t = \arg\min_z (z(g^{u_t}_{<t} + \mu_t) + \eta^{-1}(1-z)^2)$;
function $\text{Update}_{\text{weak}}\big(S_t = (g^u_{<t})_{u \in \mathcal{U}}, u_t, \mu_t, \ell_t\big)$
    $g^{u_t}_{<t+1} \leftarrow g^{u_t}_{<t} + \ell_t$;
    for $u \in \mathcal{U} \backslash \{u_t\}$ do
        $g^u_{<t+1} \leftarrow g^u_{<t}$;
    Return $S_{t+1} = \big(g^u_{<t+1}\big)_{u \in \mathcal{U}}$;
```

Theorem 14. *For any finite sets $\mathcal{U}$ and $\mathcal{X}$, $H \subset \mathcal{U}$, $x^* \in \mathcal{X}$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in [-1, 1]$, $\textsc{Collab}(\textsc{Filter}_{\text{weak}})$ with losses ℓ_t and predicted losses μ_t satisfies:*

$$\sum_{\substack{t<T \\ u_t \in H}} \ell_t \cdot p_t \leq \sum_{\substack{t<T \\ u_t \in H}} \ell_t(x^*) + \eta T + \mathcal{O}\big(\eta^{-1}(|\mathcal{U} \backslash H| + \log |\mathcal{X}|)\big)$$

Filtering with relatedness information

Now consider the setting of Section 3.4: we have a graph E, perhaps representing relationships on a social network, such that dishonest users are unlikely to be friends with honest users. We will assume that E is k-regular since this makes it easier to state our results, but this assumption is not essential.

Given a set H, define

$$\alpha_E(H) = \mathbb{P}_{(u^0, u^1) \sim E}\big(u^0 \in H \wedge u^1 \notin H\big)$$

as the probability that a random pair of related users contains one user in H and one user outside of H. Note that $\alpha_E(H)$ is strictly smaller than the probability that a random relative of an honest user is dishonest, which we expect to be smaller than the fraction of dishonest users.

We can use the regularizer R_E from Section 3.4 in order to solve the filtering problem with a regret that depends on $\alpha_E(H)$.

Lemma 12. *Fix a finite set $\mathcal{U}$ and a k-regular graph E. For any $H \subset \mathcal{U}$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in \mathbb{R}$, define $S_0 = \text{Init}_E(\mathcal{U}, \eta)$, $z_t = \text{Predict}_E(S_t, u_t, \mu_t)$, $S_{t+1} = \text{Update}_E(S_t, u_t, \mu_t, \ell_t)$.*

Algorithm 8: Filtering with relatedness information

```
$\mathcal{V} \leftarrow \{0,1\}$;
$\mathcal{M}^+ \leftarrow$ as defined in Section 3.2;
$R_E \leftarrow$ as defined in Section 3.2;
$D_u \leftarrow$ the matrix in $\mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$ with a 1 in the $(u,1)(u,1)$ entry and zeros everywhere else;
function Init$_E(\mathcal{U}, \eta)$
    $g_{<0} \leftarrow$ zero matrix in $\mathbb{R}^{(\mathcal{U}\times\mathcal{V})^2}$;
    Return $S_0 = g_{<0}$;
function Predict$_{\text{filter}}(S_t = g_{<t}, u_t, \mu_t)$
    $G_t \leftarrow g_{<t} + \mu_t D_{u^t}$;
    $X_t \leftarrow \min_{X \in \mathcal{M}^+}(\text{Tr}(G_t X) + R_E(X))$;
    Return $z_t = (X_t)_{(u_t,1)(u_t,1)}$;
function Update$_{\text{filter}}(S_t = g_{<t}, u_t, \mu_t, \ell_t)$
    $g_{<t+1} \leftarrow g_{<t} + \ell_t D_{u^t}$;
    Return $S_{t+1} = g_{<t+1}$;
```

Then we have:

$$\sum_{t<T} z_t \ell_t \leq \sum_{\substack{t<T \\ u_t \in H}} \ell_t + \eta \sum_{t<T} z_t (\ell_t - \mu_t)^2 + \mathcal{O}\left(\eta^{-1} |\mathcal{U}| \left(\alpha_E(H) + \frac{\log k}{k}\right)\right)$$

That is, Algorithm 6 is weakly competitive (satisfies Equation 4.1) with:

$$\mathcal{R}(H) = |\mathcal{U}| \left(\alpha_E(H) + \frac{\log k}{k}\right).$$

Proof. Let $\chi_H(u)$ be 1 if $u \in H$ and 0 otherwise.

Write $g_t = \ell_t D_{u^t}$, and $M_t = \mu_t D_{u^t}$.

We have $\sum_{t<T} z_t \ell_t = \sum_{t<T} g_t \cdot X_t$, and X_t is produced by OMD with the regularizer R_E. So we can apply Theorem 10 to bound this loss.

We can compute $\|g_t - M_t\|_{1,\infty}^2 = (\ell_t - \mu_t)^2$. Moreover, if we define X^* to be the matrix corresponding to the labeling $\mathbf{v}(u) = \chi_H(u)$, then we have $1 - \frac{1}{Nk}\text{Tr}(A_E X^*) = \alpha_E(H)$.

Thus we obtain:

$$\sum_{t<T} z_t \ell_t \leq g_{<T} \cdot X^* + \eta \sum_{t<T} (\ell_t - \mu_t)^2 + \mathcal{O}\left(\eta^{-1} |\mathcal{U}| \left(\alpha_E(H) + \frac{\log k}{k}\right)\right),$$

as desired. □

Theorem 15. *For any finite sets $\mathcal{U}$ and $\mathcal{X}$, any k-regular graph $E \subset \mathcal{U} \times \mathcal{U}$, $H \subset \mathcal{U}$, $x^* \in \mathcal{X}$, $T > 0$, $\eta > 0$, $\ell_t, \mu_t \in [-1, 1]$,* Collab(Filter$_E$) *with losses ℓ_t and predicted losses*

μ_t satisfies:

$$\sum_{\substack{t<T\\u_t\in H}} \ell_t \cdot p_t \leq \sum_{\substack{t<T\\u_t\in H}} \ell_t(x^*) + \eta T + \mathcal{O}\left(\eta^{-1}\left(|\mathcal{U}|\left(\alpha_E(H) + \frac{\log k}{k}\right) + \log|\mathcal{X}|\right)\right)$$

4.4 Contextual bandits

To extend our results to the contextual bandits setting, we start with a standard technique [4] for constructing an unbiased estimator of an arm's loss: if we don't choose a particular arm, we estimate its loss as 0; if we do choose an arm, we scale the observed loss by the inverse probability of choosing it. Though this estimator is unbiased it can have very high variance. In some sense, the chief difficulty of bandit problems is coping with this additional variance. In our setting, we handle this by making an explicit second-order correction to the loss of each expert, splitting the cost out amongst all experts rather than concentrating it on those who made high-variance recommendations.

For convenience we work with losses in $[0, 1]$ instead of $[-1, 1]$; it is straightforward to translate and rescale between these settings, since our regret bounds are "zeroth order," i.e. don't depend on the actual magnitude of the losses.

Theorem 16. *Fix any $T > 0$, $\eta \in [0, 1/2]$, $x \in \mathcal{X}$, $H \subset \mathcal{U}$, $x^* \in \mathcal{X}$, $\ell_t : \mathcal{A} \to [0, 1]$, $q_t^x \in \Delta(\mathcal{A})$, and an algorithm* Filter *satisfying Equation 4.1. Then Algorithm 9 with recommendations q_t^x and losses ℓ_t satisfies (in expectation):*

$$\sum_{\substack{t<T\\u_t\in H}} \ell_t(a_t) \leq \sum_{\substack{t<T\\u_t\in H}} \ell_t \cdot q_t^{x^*} + \mathcal{O}\left(\eta S T + \frac{\log|\mathcal{X}| + \mathcal{R}(H)}{\eta}\right).$$

In contrast with Theorem 12, this result requires that Filter be strongly competitive yet obtains a bound in terms of the total number of rounds T (rather than the number involving a user in H).

Proof. In expectation, we have

$$\begin{aligned} r_t(x) &= \sum_a q_t(a)\left(\frac{q_t^x(a)}{q_t(a)} - 1\right)\ell_t(a) \\ &= \sum_a (q_t^x(a) - q_t(a))\ell_t(a) \\ &= \ell_t \cdot q_t^x - \ell_t \cdot q_t \end{aligned}$$

So it is sufficient to bound $-\sum r_t(x^*)$. We will do this in two steps.

Algorithm 9: Collaborative contextual bandits

```
$S_0 \leftarrow \text{Init}_{\text{clip}}(\mathcal{X}, \text{uniform}, \eta)$;
for $x \in \mathcal{X}$ do
    $S_0^x \leftarrow \text{Init}_{\text{filter}}(\mathcal{U}, \eta)$;
for $t = 0, 1, 2, \ldots$ do
    Observe $u_t \in \mathcal{U}$;
    for $x \in \mathcal{X}$ do
        Observe $q_t^x \in \Delta(\mathcal{A})$;
        $z_t(x) \leftarrow \text{Predict}_{\text{filter}}(S_t^x, u_t, \mathbf{0})$;
    $p_t \leftarrow \text{Predict}_{\text{clip}}(w_t, z_t, \mathbf{0})$;
    $q_t(a) \leftarrow \sum_x p_t(x) q_t^x(a)$;
    Play $a_t$ sampled from distribution $q_t$;
    Observe $\ell_t(a_t) \in [0, 1]$;
    for $x \in \mathcal{X}$ do
        $r_t(x) \leftarrow \left(\frac{q_t^x(a_t)}{q_t(a_t)} - 1\right) \ell_t(a_t)$;
        $\widetilde{r}_t(x) \leftarrow \min\{r_t(x), 1/\eta\}$;
        $\delta_\eta(r_t(x)) \leftarrow r_t(x) \widetilde{r}_t(x)$;
    for $x \in \mathcal{X}$ do
        $\hat{\ell}_t(x) \leftarrow r_t(x) - \eta \delta_\eta(r_t(x))$;
        $\ell_t^x \leftarrow \widetilde{r}_t(x)$;
        $S_{t+1}^x \leftarrow \text{Update}_{\text{filter}}(S_t^x, u_t, \mathbf{0}, \ell_t^x)$;
    $w_{t+1} \leftarrow \text{Update}_{\text{clip}}\left(w_t, z_t, \mathbf{0}, \hat{\ell}_t\right)$;
```

By applying Equation 4.1 to S^{x^*}, we have:

$$\begin{aligned}
\sum_{t<T} z_t(x^*) \widetilde{r}_t(x^*) &\leq \sum_{\substack{t<T \\ u_t \in H}} \widetilde{r}_t(x^*) + \eta \sum_{t<T} z_t(x^*) \widetilde{r}_t(x^*)^2 + \frac{\mathcal{R}(H)}{\eta} \\
\sum_{t<T} z_t(x^*) r_t(x^*) &\leq \sum_{\substack{t<T \\ u_t \in H}} r_t(x^*) + \eta \sum_{t<T} z_t(x^*) \delta_\eta(r_t(x^*)) + \frac{\mathcal{R}(H)}{\eta} \\
-\sum_{\substack{t<T \\ u_t \in H}} r_t(x^*) &\leq -\sum_{t<T} z_t(x^*) r_t(x^*) + \eta \sum_{t<T} z_t(x^*) \delta_\eta(r_t(x^*)) + \frac{\mathcal{R}(H)}{\eta} \qquad (4.3)
\end{aligned}$$

where the second line follows by adding $\sum_{t<T} z_t(x^*)(\widetilde{r}_t(x^*) - r_t(x^*))$ to both sides and using the fact that if $\widetilde{r}_t(x^*) < r_t(x^*)$ then $\eta \widetilde{r}_t(x^*) = 1$ (as well as the fact that $\widetilde{r}_t(x^*) \leq r_t(x^*)$).

We have:

$$\begin{aligned}
\sum_x p_t(x)\delta_\eta(r_t(x)) &\leq \sum_x p_t(x)r_t(x)^2 \\
\sum_x p_t(x)r_t(x)^2 &= \sum_{x,a} p_t(x)\frac{(q_t^x(a)-q_t(a))^2}{q_t(a)}\ell_t(a_t)^2 \\
&\leq \sum_{x,a} p_t(x)\frac{(q_t^x(a)-q_t(a))^2}{q_t(a)} \\
&= \sum_{x,a} p_t(x)\left(\frac{q_t^x(a)^2}{q_t(a)} - 2q_t^x(a) + q_t(a)\right) \\
&\leq \sum_a \left(\max_{x\in\mathcal{X}} q_t^x(a) - 2q_t(a) + q_t(a)\right) \\
&\leq S-1
\end{aligned}$$

We compute:

$$\begin{aligned}
r_t \cdot p_t &= \sum_x p_t(x)r_t(x) \\
&= \sum_x p_t(x)\left(\frac{q_t^x(a_t)}{q_t(a_t)} - 1\right)\ell_t(a_t) \\
&= \left(\frac{q_t(a_t)}{q_t(a_t)} - \sum_x p_t(x)\right)\ell_t(a_t) \\
&= 0
\end{aligned}$$

Thus

$$\begin{aligned}
\widehat{\ell}_t \cdot p_t &= r_t \cdot p_t - \eta\sum_x p_t(x)\delta_\eta(r_t(x)) \\
&= -\eta\sum_x p_t(x)\delta_\eta(r_t(x)) \\
&> -\eta S
\end{aligned}$$

Using the fact that $\eta\delta_\eta(r_t(x)) \leq |r_t(x)|$, we have:

$$\begin{aligned}
\sum_x p_t(x)z_t(x)\widehat{\ell}_t(x)^2 &= \sum_x p_t(x)z_t(x)(r_t(x) - \eta\delta_\eta(r_t(x)))^2 \\
&\leq 4\sum_x p_t(x)z_t(x)r_t(x)^2 \\
&\leq 4S
\end{aligned}$$

If $r_t(x) \geq 0$, then $\eta\delta_\eta(r_t(x)) \leq r_t(x)$, and so $r_t(x) - \eta\delta_\eta(r_t(x)) \geq 0$. If $r_t(x) < 0$, note that $r_t(x) \geq -\ell_t(a_t) \geq -1$, and we have $\eta\delta_\eta(r_t(x)) \leq \eta r_t(x)^2 < 1$, so $r_t(x) - \eta\delta_\eta(r_t(x)) \geq -2$. In either case, $\eta\widehat{\ell}_t(x) \geq -1$. Thus we can apply Theorem 6 to S_t and the expert x^*:

$$\begin{aligned}
\sum_{t<T} z_t(x^*)\widehat{\ell}_t \cdot p_t &\leq \sum_{t<T} z_t(x^*)\widehat{\ell}_t(x^*) + \eta \sum_{t<T}\sum_{x} p_t(x)z_t(x)\widehat{\ell}_t(x)^2 + \frac{\log|\mathcal{X}|}{\eta} \\
-\eta \sum_{t<T} z_t(x^*)S &\leq \sum_{t<T} z_t(x^*)\widehat{\ell}_t(x^*) + 4\eta ST + \frac{\log|\mathcal{X}|}{\eta} \\
0 &\leq \mathcal{O}(\eta ST) + \sum_{t<T} z_t(x^*)\widehat{\ell}_t(x^*) + \frac{\log|\mathcal{X}|}{\eta} \\
0 &\leq \mathcal{O}(\eta ST) + \sum_{t<T} z_t(x^*)r_t(x^*) - \eta\sum_{t<T} z_t(x^*)\delta_\eta(r_t(x^*)) + \frac{\log|\mathcal{X}|}{\eta} \\
-\sum_{t<T} z_t(x^*)r_t(x^*) + \eta\sum_{t<T} z_t(x^*)\delta_\eta(r_t(x^*)) &\leq \mathcal{O}(\eta ST) + \frac{\log|\mathcal{X}|}{\eta} \qquad (4.4)
\end{aligned}$$

Combining with Equation 4.3, we obtain:

$$\begin{aligned}
-\sum_{\substack{t<T\\u_t\in H}} r_t(x^*) &\leq -\sum_{t<T} z_t(x^*)r_t(x^*) + \eta\sum_{t<T} z_t(x^*)\delta_\eta(r_t(x^*)) + \frac{\mathcal{R}(H)}{\eta} \\
&\leq \mathcal{O}(\eta ST) + \frac{\log|\mathcal{X}|}{\eta} + \frac{\mathcal{R}(H)}{\eta}
\end{aligned}$$

as desired.

□

Lower bound

Theorem 12 satisfies a regret bound of the form $\eta ST + \eta^{-1}N$. This bound does not become non-trivial until $T > NS$. If S is small this bound can be meaningful—for example it is an attractive bound when $\mathcal{X}$ is large but $|\mathcal{A}| = \mathcal{O}(1)$. But when S is large—for example, in the bandits setting where $S = M$—this bound is essentially as bad as having each user independently solve the bandits problem. In this section we prove that this limitation is inherent, and that there are essentially no robust collaborative learning algorithms for the (non-contextual) bandits problem.

Theorem 17. *Fix sets $\mathcal{U}$ and $\mathcal{A} = \mathcal{X}$ of sizes N and S respectively. Let $T = NS/2$. There is an (adaptive) sequence of losses ℓ_t such that for any algorithm, there is an (adaptively chosen) set H and arm a such that:*

$$\sum_{\substack{t<T\\u_t\in H}} \ell_t(a) = 0$$

but in expectation

$$\sum_{\substack{t<T \\ u_t \in H}} \ell_t(a_t) = \Omega(T).$$

Proof. Choose u_t so that each user participates in $S/2$ of the first $T = NS/2$ rounds.

For each pair $(a, u) \in \mathcal{A} \times \mathcal{U}$, set $\ell(a, u)$ to be 1 with probability 1/2 and 0 with probability 1/2. Say that an arm a is *fresh* in round t if there is no $t' < t$ with $a_{t'} = a$ and $u_{t'} = u_t$. We set $\ell_t(a) = \ell(a, u_t)$ if a is fresh in round t, otherwise we set $\ell_t(a) = 0$.

It is easy to verify that when we output a fresh arm a_t our expected loss is 1/2, since $\ell(a_t, u_t)$ is independent of everything observed by the algorithm so far. And when we output an arm that isn't fresh, our expected loss is 1.

On the other hand, it is easy to verify that there is an arm a^* which is pulled by at most half of the users. Of the half of users who never pull a^*, with high probability an $\Omega(1)$ fraction will have $\ell(a^*, u) = 0$. Take H to be the set of users who never pulled a^* and for whom $\ell(a^*, u) = 0$. This set has $\sum_{\substack{t<T \\ u_t \in H}} \ell_t(a^*) = 0$.

There is one remaining subtlety—namely, although our expected loss in each round is at least 1/2 regardless of what decisions we make, the expected loss conditioned on $u_t \in H$ might be positive. However, by the Chernoff bound, in expectation $o(N)$ users receive a total loss below $\frac{1}{2}S - S^{2/3}$. Meanwhile, $|H| = \Omega(N)$. Thus even if H consists of those users with the smallest losses, it contains in expectation $\Omega(N)$ users whose loss is $\frac{1}{2}S - S^{2/3} = \Omega(S)$. Thus in expectation $\sum_{\substack{t<T \\ u_t \in H}} \ell_t(a_t) = \Omega(NS)$. □

Chapter 5

Collaborative filtering

So far, we have considered learning problems in which users must identify a single expert $x \in \mathcal{X}$. In reality, we often need to learn about the quality or reliability of a large number of separate resources. For example, we may want to determine which merchants are honest, which movies are worth watching, or which peers are trustworthy. Discovering the single most trustworthy peer is not especially helpful—we want to learn to decide, for any given peer, whether they are trustworthy enough to interact with.

We model this situation as a sequence of go/no-go decisions: in round t, user u_t^0 is given a resource x_t (or another user u_t^1), and must decide whether to interact or not interact. If they decide interact they receive a loss $\ell_t \in [-1, 1]$, if they don't they receive a loss of 0. For example, a round may be initiated when a user finds a merchant in an online marketplace, and needs to decide whether to purchase an item from that merchant[1]

In these problems, collaborative algorithms are especially important—data from other users is the most important resource that we have, since we may never have encountered a particular merchant or peer. On the other hand, there are especially simple manipulative strategies. A dishonest merchant may simply be able to pay for dishonest reviews; even if a small minority of users are willing to take the offer, it may easily be enough to swamp the honest users.

To solve the collaborative filtering problem, we must address two key technical challenges. The first challenge is the computational complexity of optimizing over an exponential space of possible filtering policies. This is addressed by applying online local learning (from Chapter 3)[2]. The second challenge is to achieve low regret for every set of users simultaneously. This can be achieved in a computationally inefficient way using existing results in the sleeping experts setting [19, 8] (or by simply applying the results from the previous chapter with an exponentially large space of experts). Our key contribution is to show how to combine

[1] Incidentally, this is precisely the collaborative version of the problem that we introduced and solved in the previous chapter.

[2] We also applied this technique to solve the non-collaborative filtering problem in the last chapter, but in this chapter we will have to apply it in a different way.

these ideas into an algorithm which is simultaneously efficient and robust to manipulative behavior[3].

5.1 Model

Users and resources

We take as given a finite set $\mathcal{U}$ of users and a finite set $\mathcal{X}$ of *resources* such as blog posts, hotels, or merchants.

In each round $t = 1, 2, \ldots$, we are given a user $u_t \in \mathcal{U}$ and a resource $x_t \in \mathcal{X}$, and nature fixes some loss $\ell_t \in [-1, 1]$ that we will obtain if we choose to interact. We then pick $z_t \in \{0, 1\}$, potentially stochastically, indicating whether an interaction should occur. If we pick $z_t = 1$, then we observe the loss ℓ_t, and user u_t incurs the loss ℓ_t.

For any set of users $H \subset \mathcal{U}$, define $\ell^0_{<T}(H)$ to be the total loss:

$$\ell^0_{<T}(H) = \sum_{\substack{t<T \\ u_t \in H}} z_t \ell_t.$$

Define $\mathrm{OPT}_{<T}(H)$ to be the total loss that users in H would have obtained over the first T rounds by choosing the optimal set of resources S and interacting only with resources from that set. That is, define:

$$\mathrm{OPT}_{<T}(H) = \min_{S \subset \mathcal{X}} \sum_{\substack{t \leq T \\ u_t \in H \\ x_t \in S}} \ell_t.$$

The *regret* of the users in H is the difference between the benchmark $\mathrm{OPT}_{<T}(H)$ and their actual loss $\ell_{<T}(H)$.

Interactions between users

We are often interested in interactions *between two users* u^0_t and u^1_t rather than between a user and a static resource. In this chapter, we will consider the case where the losses are symmetric[4]: in round t we observe two users $u^0_t, u^1_t \in \mathcal{U}$, nature picks $\ell_t \in [-1, 1]$, we output $z_t \in \{0, 1\}$, and if we output $z_t = 1$ then *both users* receive a loss of ℓ_t.

In this case, we define the loss:

$$\ell^*_{<T}(H) = \sum_{\substack{t<T \\ u^0_t \in H}} z_t \ell_t + \sum_{\substack{t<T \\ u^1_t \in H}} z_t \ell_t.$$

[3] An earlier version of this result appeared in *Provably Manipulation-Resistant Reputation Systems* in COLT 2016.

[4] It is possible to reduce the general case to the symmetric case, under certain *ex ante* symmetry conditions, by introducing a currency for keeping track of obligations and using this currency to equalize the effective losses of different users. This is discussed in [11] but will not be covered in this chapter.

Our benchmark is the payoff that users in H would have obtained by interacting with each other and only with each other:

$$\mathrm{OPT}^*_{<T}(H) = \sum_{\substack{t<T \\ u_t^0, u_t^1 \in H}} \ell_t,$$

that is, our goal is for $\ell^*_{<T}(H)$ to be close to $2 * \mathrm{OPT}^*_{<T}(H)$ for every set H.

Unifying the models

We will work in a single formalism which captures both interactions amongst users, and between users and resources.

First, note that in the users+resources model, we can take $\mathcal{U} = \mathcal{X} = \mathcal{U} \cup \mathcal{X}$ without loss of generality: there will simply be many "resources" that never appear as x_t, and many "users" that never appear as u_t. Then in each round we are given $u_t^0, u_t^1 \in \mathcal{U}$, with u_t^0 representing a user and u_t^1 representing a resource. Moreover, we can consider a set H which contains both the honest users, and the optimal set of resources for them to interact with. Thus if we take the maximum of $\mathrm{OPT}^*_{<T}(H)$ over all sets H containing precisely the honest users, we obtain exactly $\mathrm{OPT}_{<T}(H)$. In particular, an algorithm which has low regret against $\mathrm{OPT}^*_{<T}(H)$ for every set H will also have low regret against $\mathrm{OPT}_{<T}(H)$.

So the only difference between our models is whether we care about the payoff $\ell_{<T}(H)$ or $\ell^*_{<T}(H)$, i.e. do we care only about the payoff in rounds where $u_t^0 \in H$, or do we also care about the payoff in rounds where $u_t^1 \in H$? (There is also a factor of 2 in the definition of the benchmark.)

We can handle both of these situations at once by considering two different losses:

$$\ell^i_{<T}(H) = \sum_{\substack{t<T \\ u_t^i \in H}} z_t \ell_t.$$

We will design an algorithm for which $\ell^i_{<T}(H)$ is close to $\mathrm{OPT}^*_{<T}(H)$ for both $i = 0, 1$. By applying this bound with $i = 0$ we prove that $\ell_{<T}(H)$ is close to $\mathrm{OPT}_{<T}(H)$, and by summing up the bounds for $i = 0$ and $i = 1$ we prove that $\ell^*_{<T}(H)$ is close to $2 * \mathrm{OPT}^*_{<T}(H)$.

5.2 Our algorithm

Our first attempt is to apply online learning to maximize the total welfare of all users, by viewing our problem as a contextual bandits problem. The set of actions is $\mathcal{A} = \{0, 1\}$, and the set of experts is $\mathcal{X} = \{\mathbf{v} : \mathcal{U} \to \{0, 1\}\}$. The recommendation of expert $\mathbf{v}$ in round t is the action $\mathbf{v}(u_t^0)\mathbf{v}(u_t^1)$. The payoff of action s in round t is $\ell_t \cdot s$. This will end up being the core of our approach, but it has two fundamental problems:

1. There are exponentially many maps $\mathbf{v} : \mathcal{U} \to \{0, 1\}$, and so this algorithm is intractable.

2. This algorithm guarantees that the total payoff of *all* users is close to the optimum, but it doesn't make any guarantee for the *honest* players. That is, we can bound $\sum_{t<T} z_t\ell_t$, but not $\sum_{\substack{t<T \\ u_t^i \in H}} z_t\ell_t$.

The first problem can be solved by using online local learning, as described in Chapter 3 The second problem can be solved using algorithms based on specialists [13] or time selection functions [8].

Unfortunately, it is not clear how to combine the idea of specialists with online local learning. Our main contribution is to combine these two ideas, to obtain an efficient algorithm which simultaneously achieves a tight regret bound for every subset H.

To implement this idea, we consider the set of labelings $\mathbf{v} : \mathcal{U} \to \{-1, 0, 1\}$. We could consider these labelings as strategies, as discussed in the last section. Instead, we will view a labeling as a *modification* to a strategy, as follows:

1. If $\mathbf{v}(u_t^0) \neq 0$ and $\mathbf{v}(u_t^1) \neq 0$, output $z_t = 1$.
2. If $\mathbf{v}(u_t^0) = 0$ and $\mathbf{v}(u_t^1) = 1$, output $z_t = 0$.
3. If $\mathbf{v}(u_t^0) = -1$ and $\mathbf{v}(u_t^0) = 0$, output $z_t = 0$.
4. Otherwise, defer to the current strategy

Given a distribution $p \in \Delta\big(\{-1, 0, 1\}^2\big)$, define

$$\begin{aligned} J(p) &= p(-1,-1) + p(-1,1) + p(1,-1) + p(1,1) \\ C(p) &= p(-1,0) + p(0,1) \end{aligned}$$

Then the rules above suggest an update function U which takes as input a distribution p over pairs of labels and a probability q of outputting $z_t = 1$, and outputs a new probability $U(p,q)$ of outputting $z_t = 1$:

$$U(p,q) = J(p) + (1 - J(p) - C(p))q$$

This map has a fixed point:

$$q^*(p) = \frac{J(p)}{J(p) + C(p)}$$

That is, we can compute:

$$\begin{aligned} U(p, q^*(p)) &= J(p) + (1 - J(p) - C(p))\frac{J(p)}{J(p) + C(p)} \\ &= J(p) + \frac{J(p)}{J(p) + C(p)} - (J(p) + C(p))\frac{J(p)}{J(p) + C(p)} \\ &= J(p) + q^*(p) - J(p) \\ &= q^*(p). \end{aligned}$$

Our algorithm will be to use online local learning to get a distribution p_t over pairs of labels for u_t^0, u_t^1, and to output $z_t = 1$ with probability $q^*(p_t)$.

The point of using the fixed point is to ensure that the modifications corresponding to labels output by online local learning do not improve the total loss (they can't improve the loss, since they have no effect whatsoever). But by the regret bound of online local learning, if these modifications aren't an improvement, then no fixed labeling $\mathbf{v}$ would result in an improvement (a similar idea is used in [8] to bound the swap regret). And this in turn implies that no set of users H could do better by only interacting with each other.

Algorithm 10: CollabFilter(R)

```
$S_0 \leftarrow \mathrm{Init}_{\mathrm{local}}(\mathcal{U}, \{-1,0,1\}, \eta^{-2}R)$;
for $t = 0, 1, \dots$ do
    Observe $u_t^0, u_t^1 \in \mathcal{U}$;
    $p_t \leftarrow \mathrm{Predict}_{\mathrm{local}}(S_t, u_t^0, u_t^1, \mathbf{0})$;
    $q_t \leftarrow q^*(p_t)$;
    $\widehat{q_t} \leftarrow (1-\eta)q_t + \eta$;
    Output $z_t = 1$ with probability $\widehat{q_t}$;
    Observe $z_t \cdot \ell_t$;
    $\widehat{\ell_t}(p) \leftarrow \frac{z_t \cdot \ell_t}{\widehat{q_t}}(J(p) - q_t J(p) - q_t C(p))$;
    $S_{t+1} \leftarrow \mathrm{Update}_{\mathrm{local}}\left(S_t, u_t^0, u_t^1, \mathbf{0}, \widehat{\ell_t}\right)$;
```

Theorem 18. *Suppose that* LocalLearning(R) *satisfies Equation 3.1. For each* $i \in \{0,1\}, H \subset \mathcal{U}, T > 0, \eta > 0$, *Algorithm 10 satisfies*

$$\ell^i_{<T}(H) \leq \mathrm{OPT}_{<T}(H) + \mathcal{O}\left(\eta \sum_{t<T} \left(\ell_t^2 + |\ell_t|\right) + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2}\right)$$

where $\mathbf{v}_H$ *is the map that assigns* $(-1)^{i+1}$ *to all elements of* H *and* 0 *to all elements* $\mathcal{U} \backslash H$.

Proof. Note that $\mathbb{E}\left[\widehat{\ell_t}(p)\right] = \mathbb{E}[\ell_t(U(p, q_t) - q_t)]$. Thus $\widehat{\ell_t}(p_t) = 0$, since $U(p_t, q_t) = q_t$.

Let $p_t^* \in \Delta\left(\{-1,0,1\}^2\right)$ assign probability 1 to the pair $(\mathbf{v}_H(u_t^0), \mathbf{v}_H(u_t^1))$. We can compute:

$$\sum_{\substack{t<T \\ x_i^t \in H}} U(p_t^*, q_t)\ell_t = \sum_{\substack{t<T \\ u_t^0 \in H \\ u_t^1 \in H}} \ell_t = \mathrm{OPT}_{<T}(H).$$

Moreover, if $x_i^t \notin H$, $U(p_t^*, q_t) = q_t$, and so

$$\sum_{\substack{t<T \\ u_t^0 \in H}} U(p_t^*, q_t)\ell_t = \sum_{\substack{t<T \\ u_t^0 \in H}} \ell_t q_t.$$

Thus, in expectation:

$$\begin{aligned}
\sum_{\substack{t<T \\ u_t^0 \in H}} \ell_t q_t - \mathrm{OPT}_{<T}(H) &= \sum_{t<T} \ell_t (q_t - U(p_t^*, q_t)) \\
&= -\sum_{t<T} \widehat{\ell}_t(p_t^*) \\
&= \sum_{t<T} \widehat{\ell}_t(p_t) - \sum_{t<T} \widehat{\ell}_t(p_t^*) \\
&\leq \eta^2 \sum_{t<T} \left\| \widehat{\ell}_t \right\|_\infty^2 + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2} \\
&\leq \eta^2 \sum_{t<T} \ell_t^2 z_t / \widehat{q_t}^2 + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2} \\
&= \eta^2 \sum_{t<T} \ell_t^2 / \widehat{q}_t + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2} \\
&\leq \eta \sum_{t<T} \ell_t^2 + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2}
\end{aligned}$$

Finally, observe that $|q_t - \widehat{q}_t| \leq \eta$, so we have

$$\begin{aligned}
\sum_{\substack{t<T \\ u_t^0 \in H}} \ell_t \widehat{q}_t &\leq \sum_{\substack{t<T \\ u_t^0 \in H}} \ell_t q_t + \eta \sum_{t<T} |\ell_t| \\
&\leq \mathrm{OPT}_{<T}(H) + \eta \sum_{t<T} \big(\ell_t^2 + |\ell_t|\big) + \frac{\mathcal{R}(\mathbf{v}_H)}{\eta^2}
\end{aligned}$$

as desired. □

Corollary 1. *Fix any set $\mathcal{U}$. For any sequences $u_t^0, u_t^1 \in \mathcal{U}$ and $\ell_t \in [-1,1]$ and any $H \subset \mathcal{U}$ $\textsc{CollabFilter}(R_{\mathrm{logdet}})$ satisfies in expectation:*

$$\sum_{\substack{t<T \\ u_t^0 \in H}} z_t \ell_t \leq \mathrm{OPT}_{<T}(H) + \mathcal{O}\left(\eta T + \frac{N}{\eta^2}\right).$$

Proof. Apply Theorem 18 and Theorem 9. □

Corollary 2. *Fix any set $\mathcal{U}$, and let E be a k-regular graph on E. for any sequences $u_t^0, u_t^1 \in \mathcal{U}$ and $\ell_t \in [-1,1]$, and any $H \subset \mathcal{U}$, $\textsc{CollabFilter}(R_E)$ satisfies in expectation:*

$$\sum_{\substack{t<T \\ u_t^0 \in H}} z_t \ell_t \leq \mathrm{OPT}_{<T}(H) + \mathcal{O}\left(\eta T + \frac{N\big(\alpha_E(H) + \frac{\log k}{k}\big)}{\eta^2}\right),$$

where $\alpha_E = \mathbb{P}_{(u_t^0, u_t^1) \sim E}(u_t^0 \in H \wedge u_t^1 \notin H)$.

Proof. Apply Theorem 18 and Theorem 11. □

5.3 Lower bound

Theorem 19. *For any algorithm and every N and every $T > N$, there is a sequence of payoffs p_t and users u_t^0, u_t^1 and a set H such that*

$$\mathrm{OPT}_{<T}(H) > \ell^0_{<T}(H) + \Omega\left(\sqrt{NT}\right)$$

in expectation.

Proof. Fix a single user $u^* \in \mathcal{U}$. In each round we take $u_t^0 = u^*$; we take u_t^1 to rotate through the other elements of $\mathcal{U}$ such that each participate in T/N rounds. We take ℓ_t to be a zero-mean ±1 random variable. Define $H = \{u^*\} \cup \left\{u \mid \sum_{t<T:u_t=u} \ell_t < 0\right\}$. Then $\ell^0_{<T}(H) = \sum_{t<T} z_t \ell_t$, which is zero in expectation regardless of the algorithm's choices. On the other hand, in expectation $\mathrm{OPT}_{<T}(H)$ is $-N/2$ times the expectation of the absolute value of a sum of T/N zero-mean ±1 random variables, which is $\sqrt{T/N}$. Thus in expectation $\ell^0_{<T}(H) - \mathrm{OPT}_{<T}(H) = \frac{1}{2}N\sqrt{T/N} = \frac{1}{2}\sqrt{TN}$, and in particular there is some sequence of choices by nature for which this gap is obtained (in expectation over the algorithm's random choices). □

This proof deals with a degenerate case where there is a single user u^* who participates in every round. But intuitively this is the *easiest* case where our regret should be lowest, and it's easy to see that there is no way to avoid the problem by e.g. assuming that different users' participation is balanced.

Chapter 6

Conclusion

We introduced the model of manipulation-resistant online learning, which achieves strong guarantees even when a majority of the training data is controlled by adversarial manipulators; we then proposed manipulation-resistant algorithms for prediction with expert advice, contextual bandits, and a natural collaborative filtering problem.

Our algorithms are as statistically efficient as traditional algorithms when there are no malicious users, and pay only a modest additional cost for each malicious users. In particular, these costs depend on the number of malicious *users* and not on the amount of malicious data. In each of our settings, we provide the first algorithm with this property.

In our final section we discuss a range of natural questions that remain open.

Open problems

Online local learning, bandit feedback, strong competition. Our algorithm for online local learning achieves a bound that depends on $\sum_t \|\ell_t\|_\infty^2$ rather than

$$\sum_t \|\ell_t\|_p^2 = \sum_t \sum_{v^0, v^1} p_t(v^0, v^1) \ell_t(v^0, v^1)^2.$$

For the same reason, our algorithm requires full feedback–if we only observe the payoff for the labels v_t^0, v_t^1 actually chosen by the algorithm, then the performance deteriorates dramatically.

There is no statistical obstruction to generalizing online local learning to the bandit setting where only a single payoff is observed (we can view this as a contextual bandits problem with one expert per labeling $\mathbf{v} : \mathcal{U} \to \mathcal{V}$). Doing so in a computationally efficient way would have a number of benefits for our other results:

- It would allow us to use relatedness information in contextual bandit problems, by combining Theorem 16 with a strongly competitive version of Algorithm 8.
- It would allow us to improve the dependence on η in Theorem 18.

Sharing reputation across learning problems. If the same set of users $\mathcal{U}$ faces many learning problems, and if a single set H is likely to behave honestly / have similar preferences across many of those problems, then it ought to be possible to amortize the effort of learning H—the term $\frac{\mathcal{R}(H)}{\eta}$ or $\frac{\mathcal{R}(\mathbf{v})}{\eta}$—across several problems. In practice, this could radically reduce the additional costs of collaborative learning.

If we set aside computational difficulties, this can be done by considering a joint learning problem, where the space of strategies is the Cartesian product of the possible strategies on each task. The difficulty with this approach is that the space of possible strategies grows exponentially with the number of tasks.

Note that this problem subsumes our collaborative filtering problem, which can be viewed as N separate bandit problems each of which has only 2 arms.

Collaborative convex optimization. Our main results are special cases of collaborative convex optimization. Again, setting aside computational difficulties it is straightforward to solve the collaborative version of online convex optimization, but it is unclear how to do so in a computationally efficient way.

Fast and distributed collaborative filtering. Our algorithm for collaborative filtering involves solving a semidefinite program at each step, which takes time more than quadratic in the number of users. Ideally, we would be able to exploit the special structure of this program (such as its sparsity) in order to compute a solution much more quickly, e.g. by an algorithm involving random walks. If we could get a solution with nearly linear running time, we could then aim to develop a distributed algorithm, in which each user needs to do only polylogarithmic work. Such algorithms may be much more practical in certain peer-to-peer settings.

Entry costs. Our protocols bound the damage done by a malicious user. However, a large enough number of malicious users may still be able to do a great amount of damage. If it is possible to transfer value from one user to another, then it may be possible to charge each user an "entry fee," and to redistribute these fees in such a way that the addition of malicious users does *no* damage to the honest users of the system. This would be a much more satisfying guarantee.

Bibliography

[1] Jacob Abernethy, Peter Bartlett, and Alexander Rakhlin. Multitask learning with expert advice. In *Learning Theory, 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007, Proceedings*, volume 4539 of *Lecture Notes in Artificial Intelligence*, pages 484–498, Berlin, 2007. Springer.

[2] Alon, Awerbuch, Azar, and Patt-Shamir. Tell me who I am: An interactive recommendation system. In *SPAA: Annual ACM Symposium on Parallel Algorithms and Architectures*, 2006.

[3] Arora, Hazan, and Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005.

[4] Peter Auer, Nicol O Cesa-bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem, December 08 2002.

[5] Pranjal Awasthi, Moses Charikar, Kevin A. Lai, and Andrej Risteski. Label optimal regret bounds for online local learning. *CoRR*, abs/1503.02193, 2015.

[6] Awerbuch, Azar, Lotker, Patt-Shamir, and Tuttle. Collaborate with strangers to find own preferences. *MST: Mathematical Systems Theory*, 42, 2008.

[7] Awerbuch and Kleinberg. Competitive collaborative learning. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 2005.

[8] Blum and Mansour. From external to internal regret. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 2005.

[9] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. *CoRR*, abs/1611.02315, 2016.

[10] Paul Christiano. Online local learning via semidefinite programming. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2014.

[11] Paul Christiano. Provably manipulation-resistant reputation systems. *CoRR*, abs/1411.1127, 2014.

[12] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for planted clique. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:64, 2012.

[13] Freund, Schapire, Singer, and Warmuth. Using and combining predictors that specialize. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1997.

[14] Yoav Freund and Robert E. Schapire:. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. Special issue for EuroCOLT '95.

[15] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[16] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *CoRR*, abs/1204.0136, 2012.

[17] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *CoRR*, abs/1204.0136, 2012.

[18] Wouter M. Koolen, Dmitry Adamskiy, and Manfred K. Warmuth. Putting bayes to sleep. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 135–143, 2012.

[19] Wouter M. Koolen, Dmitry Adamskiy, and Manfred K. Warmuth. Putting bayes to sleep. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 135–143, 2012.

[20] Jean-Bernard Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *IPCO: 8th Integer Programming and Combinatorial Optimization Conference*, 2001.

[21] Marissa Meyer. Do not neutralize the web's endless search, July 2010. [Online; posted 14-July-2010].

[22] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254. ACM, 2008.

[23] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In ACM, editor, *STOC '08: proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17–20, 2008*, pages 245–254, pub-ACM:adr, 2008. ACM Press.

[24] Alexander Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *CoRR*, abs/1311.1869, 2013.

[25] Paul Resnick and Rahul Sami. The influence limiter: provably manipulation-resistant recommender systems. In Joseph A. Konstan, John Riedl, and Barry Smyth, editors, *RecSys*, pages 25–32. ACM, 2007.

[26] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[27] Sherali and Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIJDM: SIAM Journal on Discrete Mathematics*, 3, 1990.

[28] Jacob Steinhardt and Percy Liang. Adaptivity and optimism: An improved exponentiated gradient algorithm. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1593–1601. JMLR.org, 2014.

[29] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. Artificial Intellegence*, 2009, 2009.

[30] Haifeng Yu, Chenwei Shi, Michael Kaminsky, Phillip B. Gibbons, and Feng Xiao. DSybil: Optimal sybil-resistance for recommendation systems. In *IEEE Symposium on Security and Privacy*, pages 283–298. IEEE Computer Society, 2009.