Online Local Learning via Semidefinite Programming

Paul Christiano*

Abstract

In many online learning problems we are interested in predicting *local* information about some universe of items. For example, we may want to know whether two items are in the same cluster rather than computing an assignment of items to clusters; we may want to know which of two teams will win a game rather than computing a ranking of teams. Although finding the optimal clustering or ranking is typically intractable, it may be possible to predict the relationships between items *as well as if* you could solve the global optimization problem exactly.

Formally, we consider an online learning problem in which a learner repeatedly guesses a pair of labels $(\ell(x), \ell(y))$ and receives an adversarial payoff depending on those labels. The learner's goal is to receive a payoff as good as the best *fixed* labeling of the items. We show that a simple algorithm based on semidefinite programming can achieve asymptotically optimal regret in the case where the number of possible labels is constant, resolving an open problem posed by Hazan, Kale, and Shalev-Schwartz [10]. Our main technical contribution is a novel use and analysis of the log det regularizer, exploiting the observation that $\log\det(\Sigma + I)$ upper bounds the entropy of any distribution with covariance matrix Σ.

1 Introduction

We are often tasked with inferring the properties of items from observations of their interactions. Frequently, we are interested in these properties primarily because they can be used to make predictions about future interactions. For example, we might:

- assign documents to clusters in order to make predictions about their similarity,
- assign characteristics to users and products in order to make appropriate recommendations,
- assign personalities to individuals to predict which groups will function well,
- assign rankings to teams in order to predict the winners of a sequence of games, or so on.

In many of these contexts, we are interested in global assignments only insofar as they help us make local predictions. Even when finding a global assignment is intractable, we may still be able to make predictions *as well as if* we had found the optimal global assignment.

In Section 1.1 we present a new formalization of this class of problems. In the case where each interaction is between two items and each item has one of $\mathcal{O}(1)$ possible labels,

*UC Berkeley. Email: paulfchristiano@eecs.berkeley.edu

our formalization is a special case of the matrix learning framework described in [10]. This includes, for example, the online max-cut problem. In this special case we are able to provide the first asymptotically optimal regret bounds, reducing the convergence time from $\mathcal{O}(\log(n))$ per item to the optimal $\mathcal{O}(1)$, which may be significant in many applications (especially those, such as recommendation systems, where *per-item* regret is a relevant metric). Moreover, our algorithm and analysis are quite natural, and considerably simplify previous approaches.

1.1 Local Prediction Problems

1.1.1 2-Local Prediction

We fix a universe of items $\mathcal{U}$ and a space of possible labels $\mathcal{L}$, with $n = |\mathcal{U}|$ and $k = |\mathcal{L}|$. Let $\Delta(\mathcal{L} \times \mathcal{L})$ be the set of probability distributions over $\mathcal{L} \times \mathcal{L}$. A *2-local prediction problem* is an online learning problem where in each round $t = 0, 1, \ldots$

1. Nature presents a pair $(i_t, j_t) \in \mathcal{U} \times \mathcal{U}$.
2. The learner submits a distribution $p_t \in \Delta(\mathcal{L} \times \mathcal{L})$.
3. Nature reveals a payoff function $c_t : \mathcal{L} \times \mathcal{L} \to [-1, 1]$.
4. The learner receives

$$\text{payoff}_t(p_t) = \sum_{a,b \in \mathcal{L} \times \mathcal{L}} c_t(a,b)\, p_t(a,b).$$

We are interested in strategies $S : \mathcal{U} \times \mathcal{U} \to \Delta(\mathcal{L} \times \mathcal{L})$. For a fixed set of choices by nature, the payoff of a strategy S can be defined straightforwardly as:

$$\text{payoff}_T(S) = \sum_{t=1}^{T} \text{payoff}_t(S(i_t, j_t)).$$

We can expand this definition in the natural way to consider strategies S which depend on the past choices of nature. We will often talk about *efficient* strategies S, which are those strategies for which $S(i_t, j_t)$ can be computed from nature's past choices in time $\text{poly}(t, n, k)$.

We are particularly interested in strategies corresponding to fixed labelings. For a labeling $\ell : \mathcal{U} \to \mathcal{L}$ we can define the strategy S_ℓ via:

$$S_\ell(i, j) = (\ell(i), \ell(j)) \text{ with probability } 1.$$

We are interested in finding strategies which perform as well as the best strategy S_ℓ, so we define

$$\text{OPT}(T) = \max_{\ell : \mathcal{U} \to \mathcal{L}} \text{payoff}_T(S_\ell).$$

We prove the following theorem:

Theorem 1 *There exists an efficient strategy S for 2-local prediction that satisfies:*

$$\textit{payoff}_T(S) \geq OPT(T) - \mathcal{O}\left(\sqrt{nk^3T}\right).$$

The dependence on n and T are optimal, but the dependence on k is not (the information-theoretic limit is $\sqrt{nT\log(k)}$). Fortunately there are already interesting 2-local prediction problems with $k = \mathcal{O}(1)$, for which we provide the first asymptotically optimal regret bounds.

1.1.2 General local prediction

These definitions can be naturally generalized to r-local prediction problems for $r > 2$. In this case, nature presents an r-tuple and the learner submits a probability distribution in $\Delta(\mathcal{L}^r)$. The payoff$_t(p_t)$ are now maps from $\mathcal{L}^r \to [-1, 1]$, and strategies S_ℓ can be defined exactly analogously to the case $r = 2$.

1.2 Examples

1.2.1 Online max-cut

Perhaps the simplest interesting example of a local prediction problem is online max-cut. In each round, the learner is given a pair of vertices $i_t, j_t \in \mathcal{U}$ and is asked to output a probability distribution over "cut" and "not cut." Nature then picks one of "cut" or "not cut" and the learner's payoff is the probability they assigned to the correct value minus the probability they assigned to the incorrect value.

A cut of $\mathcal{U}$ can be understood as a strategy in this game in a natural way. The goal in the online max-cut problem is to achieve low regret compared to the best cut.

Our algorithm can be applied to this problem directly: the space of labels is $\mathcal{L} = \{+1, -1\}$ and the payoff for a pair (a, b) is simply 1 if either $a = b$ and nature chose "not cut" or $a \neq b$ and nature chose "cut" (and -1 otherwise).

Note that nothing about the max-cut problem itself encodes the fact that we are concerning ourselves with cuts, except the fact that we are trying to perform as well as the best cut.

1.2.2 Online gambling

A significantly more complicated example is the online gambling problem. In each round, the learner is given a pair of teams $i_t, j_t \in \mathcal{U}$, and is asked to output a probability distribution over which team will win in a head-to-head contest. Nature then picks a winner, and the learner's payoff is the probability they assigned to the real winner. The learner's goal is to achieve low regret compared to the best fixed ranking of the teams. (A ranking corresponds to the strategy of deterministically predicting that the higher-ranked team wins.)

This problem can be easily fit into our setting by taking $\mathcal{L} = \{1, 2, \ldots, n\}$. A ranking of the teams then corresponds to assigning each team a separate value. The payoff for a pair $\ell(i_t) = a, \ell(j_t) = b$ is 1 if either $a > b$ and team i_t won or if $b > a$ and team j_t won.

Our algorithm does not perform well on this problem because $k = \omega(1)$.

For now, we view this example as a demonstration that the local learning framework can accomomdate significant complexity in natural settings. In fact it could go much further. For example, we could assign each team a skill level and determine a team's probability of victory by the gap between their skill and their opponents'; we could include additional variables for a team's ability to play well under varying conditions; and so on.

Thus a solution to the local learning problem for general k would give a very robust solution to the online gambling problem, in that it could accommodate many natural extensions. It seems plausible that there is a general solution to the local learning problem, but in either case it seems to be a natural generalization of the online gambling problem and either a positive or negative answer would be of natural interest.

1.3 Motivation

A wide range of prediction problems are "local" in the sense we have described, and so understanding the feasibility of such problems is a natural challenge. Our approach to this problem is motivated by the rich literature on constraint satisfaction problems in theoretical computer science, and in particular by the success of semidefinite programming (SDP) techniques.

There is an intuitive connection between *finding* a single good assignment to some variables and *competing with* the best fixed assignment. In practice it seems that both of these problems occur quite frequently: sometimes we are interested in *finding* a cut of a graph, and sometimes we are simply interested in *making predictions* about whether a given pair of items lie on the same or different sides of the cut.

Of course, we would always find a cut *in order to* make predictions. Our motivating observation was that it might be possible to substantially improve performance by "cutting out the middle man" and using SDP solutions to make predictions directly. The key ingredient in this approach is finding an appropriate regularizer for SDP solutions which can play the same role as entropy in traditional inference. We apply a log det regularizer. By observing that the log det of a matrix of moments is an upper bound on the entropy of any distribution matching those moments, we are able to show that the log determinant retains the convenient properties of entropy regularization.

A final reason to be interested in local learning is that locality serves as a building block for many other natural structural assumptions. For example, circuits and graphical models are both generated by a set of local data. Understanding the interaction between locality and learnability is a natural step in probing the boundaries of learnability, and finding simple approaches to the local learning problem may help extend positive results to more difficult cases.

1.4 Relaxations

We will work extensively with a number of relaxations of $\Delta\left(\mathcal{L}^{\mathcal{U}}\right)$. In particular, we say that a matrix $A \in \mathbb{R}^{nk \times nk}$ with rows and columns indexed by pairs in $\mathcal{U} \times \mathcal{L}$ is a *pseudodistribution over labelings* if $a, b \mapsto A_{(i,a)(j,b)}$ is a valid distribution in $\Delta\left(\mathcal{L}^2\right)$ for each i, j: that is, if $A_{(i,a)(j,b)} \geq 0$ for all i, j, a, b and $\sum_{a,b} A_{(i,a)(j,b)} = 1$ for all i, j[1].

Intuitively, $A_{(i,a)(j,b)}$ represents $\mathbb{P}\left(\ell(i) = a \wedge \ell(j) = b\right)$ for a putative distribution $\mathbb{P}$, though in fact there might not be any underlying distribution that reproduces these probabilities. Write $\mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$ for the space of pseudodistributions. In order to tighten the relaxation we might additionally require that A be positive semidefinite; write $\mathrm{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$ for the space of positive semidefinite pseudodistributions. It is easy to verify that any probability distribution $\Delta\left(\mathcal{L}^{\mathcal{U}}\right)$ corresponds to a matrix in $\mathrm{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$.

[1] We might additionally require that the marginals are consistent, i.e. that for all a, i, j, k, $\sum_b A_{(i,a)(j,b)} = \sum_b A_{(i,a)(k,b)}$. But our analysis won't make any use of that condition, and so we omit it.

For any pseudodistribution A, we can define a strategy S_A by

$$S_A(i,j) = \left(a, b \mapsto A_{(i,a)(j,b)}\right).$$

We define

$$\mathrm{OPT}_{\mathrm{SDP}}(T) = \max_{A \in \mathrm{SDP}(\mathcal{L}^{\mathcal{U}})} \mathrm{payoff}_T(S_A),$$

and our main theorem will establish:

Theorem 1 *There is an efficient strategy S for 2-local prediction such that*

$$\mathit{payoff}_T(S) \geq \mathit{OPT}_{\mathit{SDP}}(T) - \mathcal{O}\left(\sqrt{nk^3T}\right),$$

and in particular

$$\mathit{payoff}_T(S) \geq \mathit{OPT}(T) - \mathcal{O}\left(\sqrt{nk^3T}\right),$$

It is easy to see that the positive-semidefiniteness constraints are necessary: without those constraints, $\mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$ just treats each of the n^2 pairs (i,j) as a separate problem, and so it is impossible to achieve regret $o\left(\sqrt{n^2T}\right)$.. Moreover, requiring consistency of the marginals does not help in general. For example, in a max-cut problem we can assume by symmetry that the marginal distribution of each item's label is uniform.

1.5 Related work

Some of the earliest work on online learning considered the problem of competing with the best strategy from an unstructured set; for example, see [3, 4, 6, 7]. This work achieved optimal regret bounds, but the approach is not directly applicable in settings where we would like to compete with an implicitly defined and exponentially large class of strategies. These protocols can be understood as instances of the general frameworks of follow the regularized leader (FTRL) or mirror descent[9], as can our algorithm.

Similar techniques have now been applied to a much broader range of problems, including many settings with large, implicit classes of strategies [2, 8, 11, 13, 16]. In most cases, these protocols are applied to settings where the corresponding offline decision problem is easy. Our work continues in this vein, aiming to compete with a large class of combinatorially defined strategies, but we are interested in the setting where optimization over the space of strategies is intractable.

Another line of work considers online learning against spaces of positive semidefinite matrices. The matrix multiplicative weights algorithm competes with the class of positive definite matrices by using an implicit von Neumann entropy regularizer [1]. In a similar vein, [12, 14, 15] compete with the class of metrics. Although the problem statement is quite different, we apply similar techniques.

The recent results of Hazan, Kale, and Shalev-Schwartz [10] in particular are closely related to our own. They consider the setting in which learners produce outputs in $[-1,1]$ and compete with the class of matrices which can be decomposed as a difference of positive semidefinite matrices with small entries and small trace. Their framework is equivalent to ours in the case of 2-local prediction for $k = \mathcal{O}(1)$. In that setting they obtain a regret bound of $\mathcal{O}\left(\sqrt{n\log(n)\,T}\right)$, which differs from our bound by a $\log(n)$ factor. This is the

difference between a *per item* convergence time of $\mathcal{O}(1)$ and $\mathcal{O}(\log(n))$, which may be quite significant in some settings. For example, this might be the difference between a single user needing to wait $\mathcal{O}(\log(n))$ time before receiving good recommendations and needing to wait $\mathcal{O}(1)$ time.

Our techniques differ from those of [10] primarily by our choice of regularizer. Like us, they work with a semidefinite relaxation for the space of labelings (though they do not describe their approach in these terms) and solve an appropriately regularized problem. Their regularizer is the von Neumann entropy, and this leads to their regret bound of $\mathcal{O}\left(\sqrt{n\log(n)T}\right)$. We are able to improve this bound by using the log determinant. Moreover, because the log determinant is a natural analog of entropy in the setting of semidefinite programming relaxations for constraint satisfaction, we are able to give a conceptually simple analysis.

The log determinant regularizer is new in this setting but has been applied before in online learning, particularly in the context of metric learning [12, 14, 15]. Our use of this regularizer appears to be based on a fundamentally different motivation, namely as an estimate of entropy given a matrix of moments. The log determinant regularizer has also appeared (with a more similar motivation) as a regularizer for relaxed inference over Markov random fields [17], though in their context the analysis is quite different. This regularizer is perhaps most common as a barrier function in semidefinite programming; there is an analogy between our approach and a path-following algorithm for semidefinite programming, though again our analysis is quite different.

2 Our algorithm

In this section we will exhibit an algorithm which achieves regret $\mathcal{O}\left(\sqrt{nk^3T}\right)$ against the class of all strategies S_A for $A \in \mathrm{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$. Because S_A includes all strategies S_ℓ for $\ell : \mathcal{U} \to \mathcal{L}$, this yields the desired result.

2.1 Follow-the-Regularized Leader

A common approach to achieving an online regret bound is *follow-the-regularized-leader* (FTRL). FTRL chooses each move using a strategy which maximizes a linear combination of *retrospective performance* and a *regularization term*. After nature reveals each payoff, FTRL recomputes the optimum of this objective function and uses it to make a decision in the next round.

If the regularization term is strongly concave, then we can show that the maximizing strategy does not change much from one round to the next. If in addition the regularization term is bounded, we can obtain a bound on the total regret.

For completeness, we will briefly describe the algorithm, following the presentation in [9]. We also give a precise statement of the bounds we need to prove.

Formally, let $\mathcal{C}$ be a convex set and let $\mathcal{R} : \mathcal{C} \to \mathbb{R}$ be a strongly concave function which will serve as a regularizer. Consider the algorithm:

Theorem 2 *[9] Suppose that $\mathcal{R}$ is bounded between 0 and M on $\mathcal{C}$. Furthermore, suppose that whenever $|\mathit{payoff}_S(A) - \mathit{payoff}_S(A')| \geq \delta$,*

$$\mathcal{R}\left(\epsilon A + (1-\epsilon)A'\right) \geq \epsilon\mathcal{R}(A) + (1-\epsilon)\mathcal{R}\left(A'\right) + \epsilon(1-\epsilon)\gamma\delta^2$$

Algorithm 1: Follow the Regularized Leader

input: A strongly concave regularizer $\mathcal{R}$, an $\eta > 0$, and $\mathcal{C}$ a convex subset of $\mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$

for $t \in \mathbb{N}$ **do**

Set

$$A = \arg\max_{A \in \mathcal{C}} \eta \sum_{s=1}^{t-1} \mathrm{payoff}_s\left(A\right) + \mathcal{R}\left(A\right)$$

Play (a, b) with probability $A_{(i_t,a)(j_t,b)}$.

Then for an appropriate choice of η the performance of algorithm 1 is within an additive $\mathcal{O}\left(\sqrt{\frac{TM}{\gamma}}\right)$ of $\max_{A \in \mathcal{C}}$ payoff$_T\left(A\right)$.

We will aim to apply this theorem with $M = nk$ and $\gamma = \frac{1}{k^2}$, leading to an algorithm with regret $\sqrt{nk^3T}$, as desired. We believe that this analysis can be tigthened to give a regret bound of $\sqrt{nkT}$ by generalizing Lemma 7 and consequently Lemma 8, but we do not have a proof (and in either case our algorithm is most likely to be of interest when $k = \mathcal{O}\left(1\right)$).

2.2 Regularizing relaxations.

In order to apply FTRL we need to choose a convex set $\mathcal{C}$ and find a regularizer $\mathcal{R}$ which has the desired properties.

One *intractable* approach would be to take $\mathcal{C}$ to be the space of distributions over labelings, and to take $\mathcal{R}$ to be the entropy H.

Definition 3 (Entropy) *For a discrete random variable X with distribution $p(x)$, the discrete entropy is $H\left(X\right) = \mathbb{E}\left[-\log\left(p(X)\right)\right]$. For a continuous random variable X over $\mathbb{R}^N$ with density $p(x)$, the differential entropy of X is $H\left(X\right) = \mathbb{E}\left[-\log\left(p(X)\right)\right]$.*

Since there are only k^n possible labelings the entropy is bounded by $n\log(k)$. Moreover, the following standard lemma shows that the entropy is strongly concave (the proof is in the appendix):

Lemma 4 *If P and Q are probability distributions with total variation distance δ, then*

$$H\left(\epsilon P + (1-\epsilon)Q\right) \geq \epsilon H\left(P\right) + (1-\epsilon)H\left(Q\right) + \epsilon(1-\epsilon)\delta^2,$$

where H is either the discrete entropy or the differential entropy.

Of course, even representing a probability distribution over assignments $\mathcal{U} \to \mathcal{L}$ is intractable. Moreover, for our application it is unnecessary: because our payoffs depend on the labels of pairs of items, we only need to represent the pairwise marginal distributions of our labeling, i.e. the probabilities $\mathbb{P}\left(\ell\left(i\right) = a \wedge \ell\left(j\right) = b\right)$. This is precisely the information encoded by some $A \in \mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$. This motivates us to instead take the set $\mathcal{C} = \mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$, and search for some regularizer that can play the same role as H.

Unfortunately, the space $\mathrm{LP}\left(\mathcal{L}^{\mathcal{U}}\right)$ is too large, and regardless of our choice of regularizer there are information-theoretic obstructions to obtaining a sub-quadratic regret bound.

However, if we take the space SDP $\left(\mathcal{L}^{\mathcal{U}}\right)$, the situation changes entirely. Although there still need not be an actual distribution which satisfies

$$\mathbb{P}\left(\ell\left(i\right)=a\wedge\ell\left(j\right)=b\right)=A_{(i,a)(j,b)},$$

we now *can* find a jointly normally distributed family of random variables $X_{(i,a)}$ such that

$$\mathbb{E}\left[X_{(i,a)}X_{(j,b)}\right]=A_{(i,a)(j,b)}.$$

Because the gaussian is the maximum entropy distribution subject to these moment constraints, its differential entropy provides a natural upper bound on the entropy of a putative probability distribution which does have moments given by A. This motivates using the entropy of this gaussian as our regularizer $\mathcal{R}$.

The differential entropy of a gaussian with covariance matrix Σ is $\mathcal{O}\left(\log\det\left(\Sigma\right)\right)$. This motivates us to explore the suitability of this function as a regularizer. In fact we smooth this function to $\log\det\left(\Sigma+\frac{1}{k}I\right)$ to account for the difference between a discrete entropy and a differential entropy (as in [17]). For convenience of normalization, we actually take $\mathcal{R}\left(A\right)=\log\det\left(kA+I\right)$.

2.3 The log-determinant regularizer

In this section write $\mathcal{C}=\mathrm{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$ and $\mathcal{R}\left(A\right)=\log\det\left(kA+I\right)$. It is well known that maximization of concave functions over $\mathrm{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$ is tractable, so it remains to show that this choice of $\mathcal{R}$ is bounded and strongly concave.

Lemma 5 *For any $A\in SDP\left(\mathcal{L}^{\mathcal{U}}\right)$, $0\leq\mathcal{R}\left(A\right)\leq nk$.*

Proof. Since $A\geq 0$, each eigenvalue of $kA+I$ is at least 1 and hence $\log\det\left(A+\frac{1}{k}I\right)\geq 0$. On the other hand, $\mathrm{Tr}\left(A\right)=n$, so $\mathrm{Tr}\left(A+\frac{1}{k}I\right)=2kn$. Subject to this trace condition, $\det\left(A+\frac{1}{k}I\right)$ is maximized if all eigenvalues are equal, i.e. $kA+I=2I$. Hence $\log\det\left(A+\frac{1}{k}I\right)\leq\log\det\left(2I\right)=nk$. ■

The strong concavity of $\log\det\left(kA+I\right)$ is equivalent to the strong concavity of $\log\det\left(\Sigma\right)$. In order to verify strong concavity of $\log\det\left(\Sigma\right)$, we utilize its characterization as the entropy of a gaussian with covariance matrix Σ and apply the strong concavity of entropy. First, we need the following standard lemma (for example, see [5]):

Lemma 6 *For any distribution X over $\mathbb{R}^{nk}$ with covariance matrix Σ, the differential entropy of X is at most $\frac{1}{2}\log\det\left(\Sigma\right)+H_0(nk)$, where H_0 is independent of X and Σ. Moreover, equality is attained if X is gaussian.*

We also need to show that gaussians which differ in one moment necessarily have large variation distance (proved in the appendix):

Lemma 7 *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be gaussians with covariance matrices Σ_1 and Σ_2. Suppose that for some i,j we have*

$$\left|\left(\Sigma_1\right)_{ij}-\left(\Sigma_2\right)_{ij}\right|\geq\delta\left(\left(\Sigma_1\right)_{ii}+\left(\Sigma_1\right)_{jj}+\left(\Sigma_2\right)_{ii}+\left(\Sigma_2\right)_{jj}\right).$$

Then the total variation distance between $\mathcal{G}_1$ and $\mathcal{G}_2$ is $\Omega\left(\delta\right)$.

Now we can prove that $\log\det(\Sigma)$ is strongly concave:

Lemma 8 *Suppose that Σ_1 and Σ_2 are as in Lemma 7. Then*

$$\log\det\left((1-\epsilon)\Sigma_1+\epsilon\Sigma_2\right) \geq (1-\epsilon)\log\det(\Sigma_1)+\epsilon\log\det(\Sigma_2)+\Omega\left(\epsilon(1-\epsilon)\delta^2\right).$$

Proof. Let $\mathcal{G}_i$ be the gaussian with covariance matrix Σ_i. Let M be the probabilistic mixture of gaussians which puts probability $(1-\epsilon)$ on $\mathcal{G}_1$ and ϵ on $\mathcal{G}_2$. By Lemma 6, $H_0(nk)+\log\det\left((1-\epsilon)\Sigma_1+\epsilon\Sigma_2\right)$ is an upper bound on the entropy of any distribution with covariance matrix $(1-\epsilon)\Sigma_1+\epsilon\Sigma_2$, and in particular on M. Moreover, we can lower bound the entropy of M by Lemma 4 and our lower bound on the variation distance between $\mathcal{G}_1$ and $\mathcal{G}_2$:

$$\begin{aligned}
\log\det\left((1-\epsilon)\Sigma_1+\epsilon\Sigma_2\right) &\geq H(M)-H_0(nk)\\
&\geq (1-\epsilon)H(\mathcal{G}_1)+\epsilon H(\mathcal{G}_2)+\Omega\left(\epsilon(1-\epsilon)\delta^2\right)-H_0(nk)\\
&\geq (1-\epsilon)\left(H(\mathcal{G}_1)-H_0(nk)\right)+\epsilon\left(H(\mathcal{G}_2)-H_0(nk)\right)+\Omega\left(\epsilon(1-\epsilon)\delta^2\right)\\
&= (1-\epsilon)\log\det(\Sigma_1)+\epsilon\log\det(\Sigma_2)+\Omega\left(\epsilon(1-\epsilon)\delta^2\right)
\end{aligned}$$

as desired. ■

Lemma 9 *If $\text{payoff}_t(A)-\text{payoff}_t(A')\geq\delta$, then*

$$\mathcal{R}\left(\epsilon A+(1-\epsilon)A'\right)\geq\epsilon\mathcal{R}(A)+(1-\epsilon)\mathcal{R}\left(A'\right)+\Omega\left(\epsilon(1-\epsilon)\frac{\delta^2}{k^2}\right)$$

Proof sketch. If $\text{payoff}_t(A)-\text{payoff}_t(A')\geq\delta$, then kA and kA' differ by at least $\frac{\delta}{k}$ in the average entry in the block corresponding to the pair (i_t,j_t). The average diagonal entry of $kA+I$ or $kA'+I$ is only 2. So there is necessarily some entry in the block corresponding to the pair (i_t,j_t) for which the conditions of Lemma 8 apply, with parameter $\frac{\delta}{8k}$. ■

Together with the analysis of FTRL, this completes our algorithm.

3 Conclusion

3.1 Further work

3.1.1 $k=\omega(1)$

Our algorithm is optimal in the range $k=\mathcal{O}(1)$, but has very bad performance for large k; similarly, the algorithm of [10] does not generally apply to large k. Solving the problem for large k would give a robust solution to online gambling problem which would also apply directly to many natural generalizations and related learning problems.

It seems quite likely that follow the regularized leader can achieve regret $\sqrt{n\log(k)T}$ using the relaxation SDP $\left(\mathcal{L}^{\mathcal{U}}\right)$, given an appropriate choice of regularizer.

In the case of $n=1$, this problem reduces to conventional learning from experts, for which an entropy regularization is suitable. Intuitively, what is needed is a way to integrate this entropy regularization (or the von Neumann entropy regularization of matrix multiplicative weights) with the $\log\det$ regularization that performs well for large n. One way of understanding this problem is as a search for a notion of entropy that applies to

arbitrary matrices $A \in \text{SDP}\left(\mathcal{L}^{\mathcal{U}}\right)$. The bound $\log\det(A+I)$ treats each indicator variable $\ell(i) = a$ separately, and so ignores the fact that for each i the events $\ell(i) = a$ are mutually exclusive. This causes it to be a conservative overestimate for the entropy, and so to yield a suboptimal regret bound.

3.1.2 $r > 2$

In contrast with the $k = \omega(1)$ case, extending these results to $r > 2$ seems likely to be extremely difficult. It may be possible to apply existing semidefinite programming hierarchies, but if so it requires maintaining more than the first r moments of a pseudodistribution, and would be a new and interesting form of evidence for the usefulness of higher levels of these hierarchies. If existing hierarchies cannot solve the problem, it seems likely that solving r-local prediction problems for $r > 2$ would require a significant conceptual developments. It is also possible that there are complexity-theoretic obstructions, but proving lower bounds under conventional assumptions seems to be out of reach for usual techniques.

Note that we can immediately apply our results to obtain regret $\mathcal{O}\left(\sqrt{n^{\lceil \frac{r}{2} \rceil} T}\right)$ in the case $k = \mathcal{O}(1)$, by assigning labels in $\mathcal{L}^{\lceil \frac{r}{2} \rceil}$ to each set of $\lceil \frac{r}{2} \rceil$ items from $\mathcal{U}$. Understanding whether any efficient algorithm can do better may shed light on learning more broadly but also on constraint satisfaction.

3.1.3 Richer structure

Local prediction problems have particularly clean structure which makes them amenable to semidefinite programming (at least in simple cases). In many applications of interest, predictions have slightly richer structure. For example, a prediction might depend on the label of item i together with the label of one additional item which is chosen adaptively based on the label of i. This would occur for example if each of n items belonged to one of k clusters, and the properties of the items were stochastic functions of the (unobserved) characteristics of the clusters that contained them. It is interesting to ask how far we can go in this direction before impossibility results kick in; even relatively modest progress might allow completely automatic inference in a wide range of natural models.

3.2 Discussion

We have introduced the family of *local prediction problems,* and left most questions concerning this model open. We have shown that the simplest local prediction problems can be solved essentially optimally by a particularly natural algorithm, providing the first asymptotically optimal regret bounds for the online max-cut problem. In our view the conceptual simplicity of our analysis is a significant strength; we take its simplicity (together with its improvement over the previous state of the art) as some evidence that our view of online local learning is a productive one, and that further developments may be possible.

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[2] Blum, Chawla, and Kalai. Static optimality and dynamic search-optimality in lists and trees. *Algorithmica*, 36, 2003.

[3] A. Blum. On-line algorithms in machine learning. *Lecture Notes in Computer Science*, 1442, 1998.

[4] Cover. Universal data compression and portfolio selection. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 1996.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[6] Dean P. Foster and Rakesh Vohra. Regret in the on-line decision problem, July 11 1997.

[7] Freund and Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS: Journal of Computer and System Sciences*, 55, 1997.

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

[9] E. Hazan. The convex optimization approach to regret minimization. Technical report, Technion – Israel Institute of Technology, Haifa, Israel, September 2009.

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

[11] Helmbold and Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27, 1997.

[12] Prateek Jain, Brian Kulis, and Inderjit Dhillon. Online linear regression using burg entropy. Technical Report CS-TR-07-08, The University of Texas at Austin, Department of Computer Sciences, February 14 2007. Mon, 28 Jan 108 21:38:56 GMT.

[13] Kalai and Vempala. Efficient algorithms for online decision problems. *JCSS: Journal of Computer and System Sciences*, 71, 2005.

[14] Brian Kulis and Peter L. Bartlett. Implicit online learning. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 575–582. Omnipress, 2010.

[15] Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. Information-theoretic semi-supervised metric learning via entropy regularization. In *ICML*. icml.cc / Omnipress, 2012.

[16] Takimoto and Warmuth. Predicting nearly as well as the best pruning of a planar decision graph. *TCS: Theoretical Computer Science*, 288, 2002.

[17] Martin J. Wainwright and Michael I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 54(6-1):2099–2109, 2006.

4 Appendix

Proof of 4. Because the differential entropy can be written as a limit of discrete entropies under partitions, it is sufficient to verify the claim for discrete entropy.

Consider an enumeration of the union of the supports of P and Q, $x_1, x_2, \ldots$. Let $p(x_i)$ be the probability that P assigns to x_i and let $q(x_i)$ be the probability that Q assigns to x_i. For $z \in \mathbb{R}$, let $H(z) = z\log(z)$. Then

$$\begin{aligned} H(\epsilon P + (1-\epsilon)Q) &= \sum H(\epsilon p(x_i) + (1-\epsilon)q(x_i)) \\ &\geq \sum \epsilon H(p(x_i)) + (1-\epsilon)H(q(x_i)) + \frac{1}{2}\epsilon(1-\epsilon)\frac{(p(x_i)-q(x_i))^2}{p(x_i)+q(x_i)} \\ &\geq \epsilon H(P) + (1-\epsilon)H(Q) + \frac{1}{4}\epsilon(1-\epsilon)\left(\sum |p(x_i)-q(x_i)|\right)^2 \\ &= \epsilon H(P) + (1-\epsilon)H(Q) + \frac{1}{4}\epsilon(1-\epsilon)\delta^2 \end{aligned}$$

where the first inequality holds because $H''(z) = \frac{1}{z}$ and the second inequality is by Cauchy-Schwarz and the fact that $\sum (p(x_i) + q(x_i)) = 2$. ■

Proof of 7. The following proof was suggested by George Lowther in response to a question by the author on `mathoverflow.net`.

Without loss of generality, assume $\left((\Sigma_1)_{ii} + (\Sigma_1)_{jj} + (\Sigma_2)_{ii} + (\Sigma_2)_{jj}\right) = 1$, so that Σ_1 and Σ_2 differ by at least δ in their i, j entry. Consider the characteristic function $\varphi_k(u) = \mathbb{E}_{x\sim\mathcal{G}_k}\left[\exp\left(iu^T x\right)\right]$. For u with real entries, $\varphi_k(u)$ is the expectation of a function with absolute value 1 on a samples drawn from $\mathcal{G}_k$, so to bound the variation distance between $\mathcal{G}_1$ and $\mathcal{G}_2$ it suffices to exhibit some u such that $|\varphi_1(u) - \varphi_2(u)| = \Omega(\delta)$. From the definition of the Gaussian we can compute

$$\varphi_k(u) = \exp\left(-\frac{1}{2}u^T\Sigma_k u\right).$$

Let e_i, e_j be the unit vectors with their non-zero entry in coordinates i, j, respectively. Let v be one of e_i, e_j, or $e_i + e_j$, write $\alpha_k = v^T\Sigma_k v$, and write $u = \frac{v}{\sqrt{\alpha_1+\alpha_2}}$. Then we have

$$\begin{aligned} |\varphi_1(u) - \varphi_2(u)| &= \left|\exp\left(-\frac{\alpha_1}{2(\alpha_1+\alpha_2)}\right) - \exp\left(-\frac{\alpha_2}{2(\alpha_1+\alpha_2)}\right)\right| \\ &= \Omega\left(\frac{|\alpha_1-\alpha_2|}{\alpha_1+\alpha_2}\right) \\ &= \Omega(|\alpha_1-\alpha_2|) \end{aligned}$$

Where the second line follows from a constant lower bound on the derivative of $\exp(x)$ in the range $[-1, 0]$, and the third line follows from our bound on $\left((\Sigma_1)_{ii} + (\Sigma_1)_{jj} + (\Sigma_2)_{ii} + (\Sigma_2)_{jj}\right)$.

So it suffices to find some $v \in \{e_i, e_j, e_i + e_j\}$ with $\left|v^T(\Sigma_1 - \Sigma_2)v\right| = \Omega(\delta)$. But note that we can write

$$\begin{aligned} \delta &\leq (\Sigma_1 - \Sigma_2)_{ij} \\ &= e_i^T(\Sigma_1 - \Sigma_2)e_j \\ &= \frac{1}{2}\left((e_i+e_j)^T(\Sigma_1-\Sigma_2)(e_i+e_j) - e_i^T(\Sigma_1-\Sigma_2)e_i - e_j^T(\Sigma_1-\Sigma_2)e_j\right) \end{aligned}$$

and so one of the terms on the right hand side of the second line must have absolute value at least $\Omega(\delta)$, as desired.

■

Proof of 9. Write $d_{ab} = k\left|A_{(i_t,a)(j_t,b)} - A'_{(i_t,a)(j_t,b)}\right|$, $x_a = kA_{(i_t,a)(i_t,a)} + kA'_{(i_t,a)(i_t,a)} + 2$, $y_b = kA_{(j_t,b)(j_t,b)} + kA'_{(j_t,b)(j_t,b)} + 2$. Note that $\sum_a x_a = \sum_b y_b = 4k$. Note that

$$\sum_{a,b} d_{ab} = k\left|\text{payoff}_t(A) - \text{payoff}_t\left(A'\right)\right| \geq k\delta.$$

Since $\sum_{a,b}(x_a+y_b) = k\sum x_a + k\sum y_b = 8k^2$, there must be some a, b with $d_{ab} \geq \frac{1}{8k}(x_a+y_b)$. The desired result then follows from Lemma 8. ■