JMLR: Workshop and Conference Proceedings vol 23 (2012) 38.1–38.13 25th Annual Conference on Learning Theory

Near-Optimal Algorithms for Online Matrix Prediction

Elad Hazan EHAZAN@IE.TECHNION.AC.IL
Technion - Israel Inst. of Tech.

Satyen Kale SCKALE@US.IBM.COM
IBM T.J. Watson Research Center

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
John S. Cohen SL in CS, The Hebrew University

Editor: Shie Mannor, Nathan Srebro, Robert C. Williamson

Abstract

In several online prediction problems of recent interest the comparison class is composed of matrices with bounded entries. For example, in the online max-cut problem, the comparison class is matrices which represent cuts of a given graph and in online gambling the comparison class is matrices which represent permutations over n teams. Another important example is online collaborative filtering in which a widely used comparison class is the set of matrices with a small trace norm. In this paper we isolate a property of matrices, which we call (β, τ)-decomposability, and derive an efficient online learning algorithm, that enjoys a regret bound of $\tilde{O}(\sqrt{\beta\,\tau\,T})$ for all problems in which the comparison class is composed of (β, τ)-decomposable matrices. By analyzing the decomposability of cut matrices, low trace-norm matrices and triangular matrices, we derive near optimal regret bounds for online max-cut, online collaborative filtering and online gambling. In particular, this resolves (in the affirmative) an open problem posed by Abernethy (2010); Kleinberg et al. (2010). Finally, we derive lower bounds for the three problems and show that our upper bounds are optimal up to logarithmic factors. In particular, our lower bound for the online collaborative filtering problem resolves another open problem posed by Shamir and Srebro (2011). [1]

1. Introduction

We consider online learning problems in which on each round the learner receives $(i_t, j_t) \in [m]\times[n]$ and should return a prediction in $[-1, 1]$. For example, in the online collaborative filtering problem, m is the number of users, n is the number of items (e.g., movies), and on each online round the learner should predict a number in $[-1, 1]$ indicating how much user $i_t \in [m]$ likes item $j_t \in [n]$. Once the learner makes the prediction, the environment responds with a loss function, $\ell_t : [-1, 1] \to \mathbb{R}$, that assesses the correctness of the learner's prediction.

A natural approach for the learner is to maintain a matrix $\mathbf{W}_t \in [-1, 1]^{m\times n}$, and to predict the corresponding entry, $W_t(i_t, j_t)$. The matrix is updated based on the loss function and the process continues.

Without further structure, the above setting is equivalent to mn independent prediction problems - one per user-item pair. However, it is usually assumed that there is a relationship between the

1. This is an extended abstract, several key proofs are missing due to space limitations. A preprint with detailed proofs appears in Hazan et al. (2012).

different matrix entries - e.g. similar users prefer similar movies. This can be modeled in the online learning setting by assuming that there is some fixed matrix $\mathbf{W}$, in a restricted class of matrices $\mathcal{W} \subseteq [-1,1]^{m\times n}$, such that the strategy which always predicts $W(i_t, j_t)$ has a small cumulative loss. A common choice for $\mathcal{W}$ in the collaborative filtering application is to be the set of matrices with a trace norm of at most τ (which intuitively requires the prediction matrix to be of low rank). As usual, rather than assuming that some $\mathbf{W} \in \mathcal{W}$ has a small cumulative loss, we require that the regret of the online learner with respect to $\mathcal{W}$ will be small. Formally, after T rounds, the regret of the learner is

$$\text{Regret} := \sum_{t=1}^{T} \ell_t(W_t(i_t, j_t)) - \min_{\mathbf{W}\in\mathcal{W}} \sum_{t=1}^{T} \ell_t(W(i_t, j_t)),$$

and we would like the regret to be as small as possible.

A natural question is what properties of $\mathcal{W}$ enables us to derive an *efficient* online learning algorithm that enjoys low regret, and how does the regret depend on the properties of $\mathcal{W}$. In this paper we define a property of matrices, called (β, τ)-decomposability, and derive an efficient online learning algorithm that enjoys a regret bound of $\tilde{O}(\sqrt{\beta\,\tau\,T})$ for any problem in which $\mathcal{W} \subset [-1,1]^{m\times n}$ and every matrix $\mathbf{W} \in \mathcal{W}$ is (β, τ)-decomposable. Roughly speaking, $\mathbf{W}$ is (β, τ)-decomposable if a symmetrization of it can be written as $\mathbf{P} - \mathbf{N}$ where both $\mathbf{P}$ and $\mathbf{N}$ are positive semidefinite, have sum of traces bounded by τ, and have diagonal elements bounded by β.

We apply this technique to three online learning problems.

1. **Online max-cut**: On each round, the learner receives a pair of graph nodes $(i, j) \in [n] \times [n]$, and should decide whether there is an edge connecting i and j. Then, the learner receives a binary feedback. The comparison class is the set of all cuts of the graph, which can be encoded as the set of matrices $\{\mathbf{W}_A : A \subset [n]\}$, where $W_A(i, j)$ indicates if (i, j) crosses the cut defined by A or not. It is possible to achieve a regret of $O(\sqrt{nT})$ for this problem by a inefficient algorithm (simply refer to each A as an expert and apply a prediction with expert advice algorithm). Our algorithm yields a nearly optimal regret bound of $O(\sqrt{n\log(n)T})$ for this problem. This is the first *efficient* algorithm that achieves near optimal regret.

2. **Online Collaborative Filtering**: We already mentioned this problem previously. We consider the comparison class $\mathcal{W} = \{\mathbf{W} \in [-1,1]^{m\times n} : \|\mathbf{W}\|_\star \le \tau\}$, where $\|\cdot\|_\star$ is the trace norm (i.e. the sum of singular values of $\mathbf{W}$). Without loss of generality assume $m \le n$. Our algorithm yields a nearly optimal regret bound of $O(\sqrt{\tau\sqrt{n}\log(n)T})$. Since for this problem one typically has $\tau = \Theta(n)$, we can rewrite the regret bound as $O(\sqrt{n^{3/2}\log(n)T})$. In contrast, a direct application of the online mirror descent framework to this problem yields a regret of $O(\sqrt{\tau^2 T}) = O(\sqrt{n^2 T})$. The latter is a trivial bound since the bound becomes meaningful only after $T \ge n^2$ rounds (which means that we saw the entire matrix).

   Recently, Cesa-Bianchi and Shamir (2011) proposed a rather different algorithm with regret bounded by $O(\tau\sqrt{n})$ for any $T \le n^2$. However, their setting is different since they assume that each entry (i, j) is seen only once. In addition, while both the runtimes of our method and their method are polynomial, the runtime of our method is significantly smaller: for $m \approx n$, each iteration of our method can be implemented in $\tilde{O}(n^3)$ time, whereas each iteration in their algorithm requires $O(n^2)$ SVD computations, thus requiring $O(n^5)$ time.

3. **Online Gambling**: On each round, the learner receives a pair of teams $(i, j) \in [n] \times [n]$, and should predict whether i is going to beat j in an upcoming matchup or vice versa. The compar-

ison class is the set of permutations over the teams, where a permutation will predict that i is going to beat j if i appears before j in the permutation. Permutations can be encoded naturally as matrices, where $W(i, j)$ is either 1 (if i appears before j in the permutation) or 0. Again, it is possible to achieve a regret of $O(\sqrt{n \log(n) T})$ by a inefficient algorithm (that simply treats each permutation as an expert). Our algorithm yields a nearly optimal regret bound of $O(\sqrt{n \log^3(n) T})$. This resolves an open problem posed in Abernethy (2010); Kleinberg et al. (2010). Achieving this kind of regret bound was widely considered *intractable*, since computing the best permutation in hindsight is exactly the **NP**-hard minimum feedback arc set problem. In fact, Kanade and Steinke (2012) tried to show computational hardness for this problem by reducing the problem of online agnostic learning of halfspaces in a restricted setting to it. This paper shows that the problem is in fact tractable.

Finally, we derive (nearly) matching lower bounds for the three problems. In particular, our lower bound for the online collaborative filtering problem implies that the sample complexity of learning matrices with bounded entries and trace norm of $\Theta(n)$ is $\Omega(n^{3/2})$. This matches an upper bound on the sample complexity derived by Shamir and Shalev-Shwartz (2011) and solves an open problem posed by Shamir and Srebro (2011).

2. Problem statements and main results

We start with the definition of (β, τ)-decomposability. For this, we first define a symmetrization operator.

Definition 1 (Symmetrization) *Given an $m \times n$* non-symmetric *matrix $\mathbf{W}$ its symmetrization is the $(m + n) \times (m + n)$ matrix:*

$$\mathsf{sym}(\mathbf{W}) := \begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{0} \end{bmatrix}.$$

If $m = n$ and $\mathbf{W}$ is symmetric, then $\mathsf{sym}(\mathbf{W}) := \mathbf{W}$.

The main property of matrices we rely on is (β, τ)-decomposability, which we define below.

Definition 2 ((β, τ)-decomposability) *An $m \times n$ matrix $\mathbf{W}$ is (β, τ)-decomposable if there exist symmetric, positive semidefinite matrices $\mathbf{P}, \mathbf{N} \in \mathbb{R}^{p \times p}$, where p is the order of $\mathsf{sym}(\mathbf{W})$, such that the following conditions hold:*

$$\begin{aligned} \mathsf{sym}(\mathbf{W}) &= \mathbf{P} - \mathbf{N}, \\ \mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) &\le \tau, \\ \forall i \in [p] : P(i,i), N(i,i) &\le \beta. \end{aligned}$$

We say that a set of matrices $\mathcal{W}$ is (β, τ)-decomposable if every matrix in $\mathcal{W}$ is (β, τ)-decomposable.

In the above, the parameter β stands for a **b**ound on the diagonal elements of $\mathbf{P}$ and $\mathbf{N}$, while the parameter τ stands for the **t**race of $\mathbf{P}$ and $\mathbf{N}$. It is easy to verify that if $\mathcal{W}$ is (β, τ)-decomposable then so is its convex hull, $\mathrm{conv}(\mathcal{W})$. Throughout this paper, we assume for technical convenience that $\beta \geq 1$.[2] In the full version of this paper (Hazan et al., 2012) we give a connection between a (β, τ)-decomposition for a rectangular matrix $\mathbf{W}$ and its max-norm and trace norm.

Our first contribution is a generic low regret algorithm for online matrix prediction with a (β, τ)-decomposable comparison class. We also assume that all the matrices in the comparison class have bounded entries. Formally, we consider the following problem.

Online Matrix Prediction

parameters: $\beta \geq 1, \tau \geq 0, G \geq 0$
input: A set of matrices, $\mathcal{W} \subset [-1, 1]^{m \times n}$, which is (β, τ)-decomposable
for $t = 1, 2, \ldots, T$
  adversary supplies a pair of indices $(i_t, j_t) \in [m] \times [n]$
  learner picks $\mathbf{W}_t \in \mathrm{conv}(\mathcal{W})$ and outputs the prediction $W_t(i_t, j_t)$
  adversary supplies a convex, G-Lipschitz, loss function $\ell_t : [-1, 1] \to \mathbb{R}$
  learner pays $\ell_t(W_t(i_t, j_t))$

Theorem 1 *Let p be the order (i.e. the number of rows or columns) of* $\mathsf{sym}(\mathbf{W})$ *for any matrix* $\mathbf{W} \in \mathcal{W}$. *There exists an efficient algorithm for Online Matrix Prediction with the regret bound*

$$\textit{Regret} \;\leq\; 2G\sqrt{\tau\beta\log(2p)T}.$$

The Online Matrix Prediction problem captures several specific problems considered in the literature, given in the next few subsections.

2.1. Online Max-Cut

Recall that on each round of online max-cut, the learner should decide whether two vertices of a graph, (i_t, j_t) are joined by an edge or not. The learner outputs a number $\hat{y}_t \in [-1, 1]$ which is to be interpreted as a randomized prediction in $\{-1, 1\}$: predict 1 with probability $\frac{1+\hat{y}_t}{2}$ and -1 with the remaining probability. The adversary then supplies the true outcome, $y_t \in \{-1, 1\}$, where $y_t = 1$ indicates the outcome "(i_t, j_t) are joined by an edge", and $y_t = -1$ the opposite outcome. The loss suffered by the learner is the absolute loss,

$$\ell_t(\hat{y}_t) \;=\; \frac{1}{2}|\hat{y}_t - y_t|,$$

which can be also interpreted as the probability that a randomized prediction according to $\hat{y}_t$ will not equal the true outcome y_t.

The comparison class is $\mathcal{W} = \{\mathbf{W}_A | A \subseteq [n]\}$, where

$$W_A(i,j) = \begin{cases} 1 & \text{if } ((i \in A) \text{ and } (j \notin A)) \text{ or } ((j \in A) \text{ and } (i \notin A)) \\ -1 & \text{otherwise.} \end{cases}$$

That is, $W_A(i, j)$ indicates if (i, j) crosses the cut defined by A or not. The following lemma (proved in Hazan et al. (2012)) formalizes the relationship of this online problem to the max-cut problem:

2. The condition $\beta \geq 1$ is not a serious restriction since for any (β, τ)-decomposition of $\mathbf{W}$, viz. $\mathsf{sym}(\mathbf{W}) = \mathbf{P} - \mathbf{N}$, we have $\beta \geq |P(i, j)|, |N(i, j)|$ for all (i, j) since $\mathbf{P}, \mathbf{N} \succeq \mathbf{0}$; and so $2\beta \geq |P(i, j) - N(i, j)| = |W(i, j)|$. Thus, if we make the reasonable assumption that there is some $\mathbf{W} \in \mathcal{W}$ with $|W(i, j)| = 1$ for some (i, j), then $\beta \geq \frac{1}{2}$ is necessary.

Lemma 1 *Consider an online sequence of loss functions $\{\ell_t\}$ as above. Let*

$$\mathbf{W}^* = \arg\min_{\mathbf{W}\in\mathcal{W}} \sum_t \ell_t(W(i_t, j_t)) \, .$$

Then $\mathbf{W}^* = \mathbf{W}_A$ for the set A that determines the max cut in the weighted graph over $[n]$ nodes whose weights are given by $w_{ij} = \sum_{t:(i_t,j_t)=(i,j)} y_t$ for every (i,j).

A regret bound of $O(\sqrt{nT})$ is attainable for this problem as follows via an exponential time algorithm: consider the set of all 2^n cuts in the graph. For each cut defined by A, consider a decision rule or "expert" that predicts according to the matrix $\mathbf{W}_A$. Standard bounds for the experts algorithm imply the $O(\sqrt{nT})$ regret bound.

A simple way to get an efficient algorithm is to replace $\mathcal{W}$ with the class of all matrices in $\{-1,1\}^{n\times n}$. This leads to n^2 different prediction tasks, each of which corresponds to the decision if there is an edge between two nodes, which is efficiently solvable. However, the regret with respect to this larger comparison class scales like $O(\sqrt{n^2 T})$.

Another popular approach for circumventing the hardness is to replace $\mathcal{W}$ with the set of matrices whose trace-norm is bounded by $\tau = n$. However, applying the online mirror descent algorithmic framework with an appropriate squared-Schatten norm regularization, as described in (Kakade et al., 2012), leads to a regret bound that again scales like $O(\sqrt{n^2 T})$.

In contrast, our Online Matrix Prediction algorithm yields an efficient solution for this problem, with a regret that scales like $\sqrt{n\log(n)T}$. The regret bound of the algorithm follows from the following lemma (proved in Section 4):

Lemma 2 *$\mathcal{W}$ is $(1, n)$-decomposable.*

Combining the above with Theorem 1 yields:

Corollary 2 *There is an efficient algorithm for the online max-cut problem with regret bounded by $2\sqrt{n\log(n)T}$.*

We prove (in (Hazan et al., 2012)) that the upper bound is near-optimal:

Theorem 3 *For any algorithm for the online max-cut problem, there is a sequence of entries (i_t, j_t) and loss functions ℓ_t for $t = 1, 2, \ldots, T$ such that the regret of the algorithm is at least $\sqrt{nT/16}$.*

2.2. Collaborative Filtering with Bounded Trace Norm

In this problem, the comparison set $\mathcal{W}$ is the following set of $m \times n$ matrices with trace norm bounded by some parameter τ:

$$\mathcal{W} \; := \; \{\mathbf{W} \in [-1,1]^{m\times n} : \; \|\mathbf{W}\|_\star \le \tau\}. \tag{1}$$

Without loss of generality we assume that $m \le n$.

As before, applying the technique of Kakade et al. (2012) gives a regret bound of $O(\sqrt{\tau^2 T})$, which leads to trivial results in the most relevant case where $\tau = \Theta(\sqrt{mn})$. We can obtain a much better result based on the following lemma (proved in Section 4):

Lemma 3 *The class $\mathcal{W}$ given in Eq. (1) is $(\sqrt{m+n}, 2\tau)$-decomposable.*

Combining the above with Theorem 1 yields:

Corollary 4 *There is an efficient algorithm for the online collaborative filtering problem with regret bounded by* $2G\sqrt{2\tau\sqrt{n+m}\log(2(m+n))T}$*, assuming that for all* t *the loss function is* G*-Lipschitz.*

This upper bound is near-optimal, as we can also show (in (Hazan et al., 2012)) the following lower bound on the regret:

Theorem 5 *For any algorithm for online collaborative filtering problem with trace norm bounded by* τ*, there is a sequence of entries* (i_t, j_t) *and* G*-Lipschitz loss functions* ℓ_t *for* $t = 1, 2, \ldots, T$ *such that the regret of the algorithm is at least* $G\sqrt{\frac{1}{2}\tau\sqrt{n}T}$*.*

In fact, the technique used to prove the above lower bound also implies a lower bound on the sample complexity of collaborative filtering in the batch setting.

Theorem 6 *The sample complexity of learning* $\mathcal{W}$ *in the batch setting, is* $\Omega(\tau\sqrt{n}/\varepsilon^2)$*. In particular, when* $\tau = \Theta(n)$*, the sample complexity is* $\Omega(n^{1.5}/\varepsilon^2)$*.*

This matches an upper bound given by Shamir and Shalev-Shwartz (2011). The question of determining the sample complexity of $\mathcal{W}$ in the batch setting has been posed as an open problem by Shamir (who conjectured that it scales like $n^{1.5}$) and Srebro (who conjectured that it scales like $n^{4/3}$).

2.3. Online gambling

In the gambling problem, we define the comparison set $\mathcal{W}$ as the following set of $n \times n$ matrices. First, for every permutation $\pi : [n] \to [n]$, define the matrix $\mathbf{W}_\pi$ as:

$$W_\pi(i,j) \;=\; \begin{cases} 1 & \text{if } \pi(i) \le \pi(j) \\ 0 & \text{otherwise.} \end{cases}$$

Then the set $\mathcal{W}$ is defined as

$$\mathcal{W} \;:=\; \{\mathbf{W}_\pi : \; \pi \text{ is a permutation of } [n]\}. \tag{2}$$

On round t, the adversary supplies a pair (i_t, j_t) with $i_t \neq j_t$, and the learner outputs as a prediction $\hat{y}_t = W_t(i_t, j_t) \in [0,1]$, where we interpret $\hat{y}_t$ as the probability that i_t will beat j_t. The adversary then supplies the true outcome, $\hat{y}_t \in \{0,1\}$, where $\hat{y}_t = 1$ indicates the outcome "i_t beats j_t", and $\hat{y}_t = 0$ the opposite outcome. The loss suffered by the learner is the absolute loss,

$$\ell_t(y_t) \;=\; |y_t - \hat{y}_t|,$$

which can be also interpreted as the probability that a randomized prediction according to $\hat{y}_t$ will not equal to the true outcome y_t.

As before, we tackle the problem by analyzing the decomposability of $\mathcal{W}$, via the following lemma (see (Hazan et al., 2012) for proof):

Lemma 4 *The class* $\mathcal{W}$ *given in Eq. (2) is* $(O(\log(n)), O(n\log(n)))$*-decomposable.*

Combining the above with Theorem 1 yields:

Corollary 7 *There is an efficient algorithm for the online gambling problem with regret bounded by* $O(\sqrt{n\log^3(n)T})$.

This upper bound is near-optimal, as Kleinberg et al. (2010) essentially prove the following lower bound on the regret:

Theorem 8 *For any algorithm for the online gambling problem, there is a sequence of entries* (i_t, j_t) *and labels* y_t*, for* $t = 1, 2, \ldots, T$*, such that the regret of the algorithm is at least* $\Omega(\sqrt{n\log(n)T})$.

3. The Algorithm for Online Matrix Prediction

In this section we prove Theorem 1 by constructing an efficient algorithm for Online Matrix Prediction and analyze its regret. We start by describing an algorithm for Online Linear Optimization (OLO) over a certain set of matrices and with a certain set of linear loss functions. We show later that the Online Matrix Prediction problem can be reduced to this online convex optimization problem.

3.1. The (β, τ, γ)-OLO problem

In this section, all matrices are in the space of real symmetric matrices of size $N \times N$, which we denote by $\mathbb{S}^{N\times N}$.

On each round of online linear optimization, the learner chooses an element from a convex set $\mathcal{K}$ and the adversary responds with a linear loss function. In our case, the convex set $\mathcal{K}$ is a subset of the set of matrices with bounded trace and diagonal values:

$$\mathcal{K} \subseteq \{\mathbf{X} \in \mathbb{S}^{N\times N} : \mathbf{X} \succeq \mathbf{0},\ \forall i \in [N] :\ X_{ii} \le \beta,\ \mathrm{Tr}(\mathbf{X}) \le \tau\}.$$

We assume for convenience that $\frac{\tau}{N}\mathbf{I} \in \mathcal{K}$. The loss function on round t is the function $\mathbf{X} \mapsto \mathbf{X} \bullet \mathbf{L}_t \stackrel{\text{def}}{=} \sum_{i,j} X(i,j)L_t(i,j)$, where $\mathbf{L}_t$ is a matrix from the following set of matrices:

$$\mathcal{L} = \{\mathbf{L} \in \mathbb{S}^{N\times N} :\ \mathbf{L}^2 \stackrel{\text{def}}{=} \mathbf{L}\mathbf{L} \text{ is a diagonal matrix s.t. } \mathrm{Tr}(\mathbf{L}^2) \le \gamma\}.$$

The requirement for L^2 to be diagonal is purely for technical reasons. We call the above setting a (β, τ, γ)-OLO problem.

As usual, we analyze the regret of the algorithm

$$\text{Regret} := \sum_{t=1}^{T} \mathbf{X}_t \bullet \mathbf{L}_t - \min_{\mathbf{X}\in\mathcal{K}} \sum_{t=1}^{T} \mathbf{X} \bullet \mathbf{L}_t\ ,$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_T$ are the predictions of the learner.

Below we describe and analyze an algorithm for the (β, γ, τ)-OLO problem. The algorithm, forms of which independently appeared in the work of Tsuda et al. (2006) and Arora and Kale (2007), performs exponentiated gradient steps followed by Bregman projections onto $\mathcal{K}$. The projection operation is defined with respect to the quantum relative entropy divergence:

$$\Delta(\mathbf{X}, \mathbf{A}) = \mathrm{Tr}(\mathbf{X}\log(\mathbf{X}) - \mathbf{X}\log(\mathbf{A}) - \mathbf{X} + \mathbf{A}).$$

Algorithm 1 Matrix Multiplicative Weights with Quantum Relative Entropy Projections

```
Input: η
Initialize X_1 = (τ/N) I.
for t = 1, 2, ..., T: do
    Play the matrix X_t.
    Obtain loss matrix L_t.
    Update X_{t+1} = arg min_{X∈K} Δ(X, exp(log(X_t) − ηL_t)).
end for
```

Algorithm 1 has the following regret bound (essentially following Tsuda et al. (2006); Arora and Kale (2007)):

Theorem 9 *Suppose η is chosen so that $\eta\|\mathbf{L}_t\| \le 1$ for all t (where $\|\mathbf{L}_t\|$ is the spectral norm of $\mathbf{L}_t$). Then*

$$\textit{Regret} \;\le\; \eta \sum_{t=1}^{T} \mathbf{X}_t \bullet \mathbf{L}_t^2 + \frac{\tau \log(N)}{\eta}.$$

Equipped with the above we are ready to prove a regret bound for (β, τ, γ)-OLO.

Theorem 10 *Assume $T \ge \frac{\tau \log(N)}{\beta}$. Then, applying Algorithm 1 with $\eta = \sqrt{\frac{\tau \log(N)}{\beta\gamma T}}$ on a (β, τ, γ)-OLO problem, yields an efficient algorithm whose regret is at most $2\sqrt{\beta\gamma\tau \log(N) T}$.*

Proof Clearly, Algorithm (1) can be implemented in polynomial time. To analyze the regret of the algorithm we rely on Theorem 9. By the definition of $\mathcal{K}$ and $\mathcal{L}$, we get that $\mathbf{X}_t \bullet \mathbf{L}_t^2 \le \beta\gamma$. Hence, the regret bound becomes

$$\text{Regret} \;\le\; \eta\beta\gamma T + \frac{\tau \log(N)}{\eta}.$$

Substituting the value of η, we get the stated regret bound. One technical condition is that the above regret bound holds as long as η is chosen small enough so that for all t, we have $\eta\|\mathbf{L}_t\| \le 1$. Now $\|\mathbf{L}_t\| \le \|\mathbf{L}_t\|_F = \sqrt{\mathrm{Tr}(\mathbf{L}_t^2)} \le \sqrt{\gamma}$. Thus, for $T \ge \frac{\tau \log(N)}{\beta}$, the technical condition is satisfied for $\eta = \sqrt{\frac{\tau \log(N)}{\beta\gamma T}}$. ■

3.2. An Algorithm for the Online Matrix Prediction Problem

In this section we describe a reduction from the Online Matrix Prediction problem (with a (β, τ)-decomposable comparison class) to a $(\beta, \tau, 4G^2)$-OLO problem with $N = 2p$. The regret bound of the derived algorithm will follow directly from Theorem 10.

We now describe the reduction. To simplify our notation, let q be m if $\mathcal{W}$ contains non-symmetric matrices and $q = 0$ otherwise. Note that the definition of sym$(\mathbf{W})$ implies that for a pair of indices $(i, j) \in [m] \times [n]$, their corresponding indices in sym$(\mathbf{W})$ are $(i, j + q)$.

Given any matrix $\mathbf{W} \in \mathcal{W}$ we embed its symmetrization sym$(\mathbf{W})$ (which has size $p \times p$) into the set of $2p \times 2p$ positive semidefinite matrices as follows. Since $\mathbf{W}$ admits a (β, τ)-decomposition, there exist $\mathbf{P}, \mathbf{N} \succeq \mathbf{0}$ such that $\mathsf{sym}(\mathbf{W}) = \mathbf{P} - \mathbf{N}$, $\mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) \leq \tau$, and for all $i \in [p]$, $P(i,i), N(i,i) \leq \beta$. The embedding of $\mathbf{W}$ in $\mathbb{S}^{2p \times 2p}$, denoted $\phi(\mathbf{W})$, is defined to be the matrix[3]

$$\phi(\mathbf{W}) \;=\; \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}.$$

It is easy to verify that $\phi(\mathbf{W})$ belongs to the convex set $\mathcal{K}$ defined below:

$$\begin{aligned} \mathcal{K} := \Big\{ \; & \mathbf{X} \in \mathbb{S}^{2p \times 2p} \;\; \text{s.t.} \\ & \mathbf{X} \succeq \mathbf{0} \\ & \forall i \in [2p] : \; X(i,i) \;\leq\; \beta \\ & \mathrm{Tr}(\mathbf{X}) \;\leq\; \tau \\ & \forall (i,j) \in [m] \times [n] : \; (X(i, j+q) - X(p+i, p+j+q)) \;\in [-1,1] \; \Big\} \end{aligned} \tag{3}$$

We shall run the OLO algorithm with the set $\mathcal{K}$. On round t, if the adversary gives the pair (i_t, j_t), then we predict

$$\hat{y}_t = X_t(i_t, j_t + q) - X_t(p + i_t, p + j_t + q)\;.$$

The last constraint defining $\mathcal{K}$ simply ensures that $\hat{y}_t \in [-1, 1]$. While this constraint makes the quantum relative entropy projection onto $\mathcal{K}$ more complex, in (Hazan et al., 2012) we show how we can leverage the knowledge of (i_t, j_t) to get a very fast implementation.

Next we describe how to choose the loss matrices $\mathbf{L}_t$ using the subderivative of ℓ_t. Given the loss function ℓ_t, let g be a subderivative of ℓ_t at $\hat{y}_t$. Since ℓ_t is convex and G-Lipschitz, we have that $|g| \leq G$. Define $\mathbf{L}_t \in \mathbb{S}^{2p \times 2p}$ as follows:

$$L_t(i,j) = \begin{cases} g & \text{if } (i,j) = (i_t, j_t + q) \text{ or } (i,j) = (j_t + q, i_t) \\ -g & \text{if } (i,j) = (p + i_t, p + j_t + q) \text{ or } (i,j) = (p + j_t + q, p + i_t) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that $\mathbf{L}_t^2$ is a diagonal matrix, whose only non-zero diagonal entries are (i_t, i_t), $(j_t + q, j_t + q)$, $(p + i_t, p + i_t)$, and $(p + j_t + q, p + j_t + q)$, all equalling g^2. Hence, $\mathrm{Tr}(\mathbf{L}_t^2) = 4g^2 \leq 4G^2$.

To summarize, the Online Matrix Prediction algorithm will be as follows:

3. Note that this mapping depends on the choice of $\mathbf{P}$ and $\mathbf{N}$ for each matrix $\mathbf{W} \in \mathcal{W}$. We make an arbitrary choice for each $\mathbf{W}$.

Algorithm 2 Matrix Multiplicative Weights for Online Matrix Prediction

Input: $\beta, \tau, G, m, n, p, q$ (see text for definitions)
Set: $\gamma = 4G^2$, $N = 2p$, $\eta = \sqrt{\frac{\tau \log(N)}{\beta\gamma T}}$
Let $\mathcal{K}$ be as defined in Eq. (3)
Initialize $\mathbf{X}_1 = \frac{\tau}{N}\mathbf{I}$.
for $t = 1, 2, \ldots, T$: **do**
  Adversary supplies a pair of indices $(i_t, j_t) \in [m] \times [n]$.
  Predict $\hat{y}_t = X_t(i_t, j_t + q) - X_t(p + i_t, p + j_t + q)$.
  Obtain loss function $\ell_t : [-1, 1] \to \mathbb{R}$ and pay $\ell_t(\hat{y}_t)$.
  Let g be a sub-derivative of ℓ_t at $\hat{y}_t$
  Let $\mathbf{L}_t$ be as defined in Eq. (4)
  Update $\mathbf{X}_{t+1} = \arg\min_{\mathbf{X} \in \mathcal{K}} \Delta(\mathbf{X},\ \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))$.
end for

To analyze the algorithm, note that for any $\mathbf{W} \in \mathcal{W}$,

$$\phi(\mathbf{W}) \bullet \mathbf{L}_t = 2g(P(i_t, j_t) - N(i_t, j_t)) = 2gW(i_t, j_t),$$

and

$$\mathbf{X}_t \bullet \mathbf{L}_t \ = \ 2g(X_t(i_t, j_t + q) - X_t(p + i_t, p + j_t + q)) \ = \ 2g\hat{y}_t.$$

So for any $\mathbf{W} \in \mathcal{W}$, we have

$$\mathbf{X}_t \bullet \mathbf{L}_t - \phi(\mathbf{W}) \bullet \mathbf{L}_t \ = \ 2g(\hat{y}_t - W(i_t, j_t)) \geq \ 2(\ell_t(\hat{y}_t) - \ell_t(W(i_t, j_t))),$$

by the convexity of $\ell_t(\cdot)$. This implies that for any $\mathbf{W} \in \mathcal{W}$,

$$\sum_{t=1}^{T} \ell_t(\hat{y}_t) - \ell_t(W(i_t, j_t)) \ \leq \ \frac{1}{2}\left[\sum_{t=1}^{T} \mathbf{X}_t \bullet \mathbf{L}_t - \phi(\mathbf{W}) \bullet \mathbf{L}_t\right] \ \leq \ \frac{1}{2} \cdot \text{Regret}_{\text{OLO}}.$$

Thus, the regret of the Online Matrix Prediction problem is at most half the regret in the $(\beta, \tau, 4G^2)$-OLO problem.

3.2.1. PROOF OF THEOREM 1

Following our reduction, we can now appeal to Theorem 10. For $T \geq \frac{\tau \log(2p)}{\beta}$, the bound of Theorem 10 applies and gives a regret bound of $2G\sqrt{\tau\beta \log(2p)T}$. For $T < \frac{\tau \log(2p)}{\beta}$, note that in any round, the regret can be at most $2G$, since the subderivatives of the loss functions are bounded in absolute value by G and the domain is $[-1, 1]$, so the regret is bounded by $2GT < 2G\sqrt{\tau\beta \log(2p)T}$ since $\beta \geq 1$. Thus, we have proved the regret bound stated in Theorem 1.

4. Decomposability Proofs

In this section we prove the decomposability results for the comparison classes corresponds to max-cut and collaborative filtering. The decomposability result for gambling is deferred to (Hazan et al., 2012). All the three decompositions we give are optimal up to constant factors.

4.1. Proof of Lemma 2 (max-cut)

We need to show that every matrix $\mathbf{W}_A \in \mathcal{W}$ admits a $(1, n)$-decomposition. We can rewrite $\mathbf{W}_A = -\mathbf{w}_A \mathbf{w}_A^\top$ where $\mathbf{w}_A \in \mathbb{R}^n$ is the vector such that

$$w_A(i) = \begin{cases} 1 & \text{if } i \in A \\ -1 & \text{otherwise.} \end{cases}$$

Since $\mathbf{W}_A$ is already symmetric, $\mathsf{sym}(\mathbf{W}_A) = \mathbf{W}_A = -\mathbf{w}_A \mathbf{w}_A^\top$. Thus we can choose $\mathbf{P} = \mathbf{0}$ and $\mathbf{N} = \mathbf{w}_A \mathbf{w}_A^\top$. These are positive semidefinite matrices with diagonals bounded by 1 and sum of traces equals to n, which concludes the proof. Since $\mathrm{Tr}(\mathbf{w}_A \mathbf{w}_A^\top) = n$, this $(1, n)$-decomposition is optimal.

4.2. Proof of Lemma 3 (collaborative filtering)

Recall that $\|\mathbf{W}\|_\star$ is the trace norm of $\mathbf{W}$, i.e. the sum of its singular values. We need to show that every matrix $\mathbf{W} \in \mathcal{W}$, i.e. an $m \times n$ matrix over $[-1, 1]$ with $\|\mathbf{W}\|_\star \le \tau$, admits a $(\sqrt{m+n}, 2\tau)$-decomposition. The $(\sqrt{m+n}, 2\tau)$-decomposition of $\mathbf{W}$ is a direct consequence of the following theorem, setting $\mathbf{Y} = \mathsf{sym}(\mathbf{W})$, with $p = m + n$, and the fact that $\|\mathsf{sym}(\mathbf{W})\|_\star = 2\|\mathbf{W}\|_\star$ (see Hazan et al. (2012)).

Theorem 11 *Let $\mathbf{Y}$ be a $p \times p$ symmetric matrix with entries in $[-1, 1]$. Then $\mathbf{Y}$ can be written as $\mathbf{Y} = \mathbf{P} - \mathbf{N}$ where $\mathbf{P}$ and $\mathbf{N}$ are both positive semidefinite matrices with diagonal entries bounded by $\sqrt{p}$, and $\mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) = \|\mathbf{Y}\|_\star$.*

Proof Let

$$\mathbf{Y} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$$

be the eigenvalue decomposition of $\mathbf{Y}$. We now show that

$$\mathbf{P} = \sum_{i\colon \lambda_i \ge 0} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top \text{ and } \mathbf{N} = \sum_{i\colon \lambda_i < 0} -\lambda_i \mathbf{v}_i \mathbf{v}_i^\top$$

satisfy the required conditions. Clearly $\mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) = \sum_i |\lambda_i| = \|\mathbf{Y}\|_\star$. Define $\mathsf{abs}(\mathbf{Y}) = \mathbf{P} + \mathbf{N} = \sum_i |\lambda_i| \mathbf{v}_i \mathbf{v}_i^\top$. Note that

$$\mathsf{abs}(\mathbf{Y})^2 \;=\; \sum_i \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^\top \;=\; \mathbf{Y}^2.$$

We now show that *all* entries (and in particular, the diagonal entries) of $\mathsf{abs}(\mathbf{Y})$ are bounded in magnitude by $\sqrt{p}$. Since $\mathbf{P}$ and $\mathbf{N}$ are both positive semidefinite, their diagonal elements must be non-negative, so we conclude that the diagonal entries of $\mathbf{P}$ and $\mathbf{N}$ are bounded by $\sqrt{p}$ as well.

Since all the entries of $\mathbf{Y}$ are bounded in magnitude by 1, it follows that all entries of $\mathbf{Y}^2$ are bounded in magnitude by p. In particular, the diagonal entries of $\mathbf{Y}^2$ are bounded by p. Since these diagonal entries are equal to the squared lengths of the rows of $\mathsf{abs}(\mathbf{Y})$, it follows that each entry of $\mathsf{abs}(\mathbf{Y})$ is bounded in magnitude by $\sqrt{p}$. ■

This decomposition is optimal up to constant factors. Consider the matrix $\mathbf{W}$ formed by taking $m = \frac{\tau}{\sqrt{n}}$ rows of an $n \times n$ Hadamard matrix. In (Hazan et al., 2012) we prove that any $(\beta, \tilde{\tau})$-decomposition of $\mathsf{sym}(\mathbf{W})$ must have $\beta\tilde{\tau} \geq \frac{1}{4}\tau\sqrt{n}$. Since the regret bound depends on the product $\beta\tilde{\tau}$, we conclude that the decomposition obtained from Theorem 11 is optimal up to a constant factor.

5. Conclusions

In recent years the FTRL (Follow The Regularized Leader) paradigm has become the method of choice for proving regret bounds for online learning problems. In several online learning problems a direct application of this paradigm has failed to give tight regret bounds due to suboptimal "convexification" of the problem. This unsatisfying situation occurred in mainstream applications, such as online collaborative filtering, but also in basic prediction settings such as the online max cut or online gambling settings.

In this paper we single out a common property of these unresolved problems: they involve *structured matrix* prediction, in the sense that the matrices involved have certain nice decompositions. We give a unified formulation for three of these structured matrix prediction problems which leads to near-optimal convexification. Applying the standard FTRL algorithm, Matrix Multiplicative Weights, now gives efficient and near optimal regret algorithms for these problems. In the process we resolve two COLT open problems. The main conclusion of this paper is that spectral analysis in matrix predictions tasks can be surprisingly powerful, even when the connection between the spectrum and the problem may not be obvious on first sight (such as in the online gambling problem).

We leave open the question of bridging the logarithmic gap between known upper and lower bounds for regret in these structured prediction problems. Note that since all the three decompositions in this paper are optimal up to constant factors, one cannot close the gap by improving the decomposition; some fundamentally different algorithm seems necessary. It would also be interesting to see more applications of the (β, τ)-decomposition for other online matrix prediction problems.

References

J. Abernethy. Can we learn to gamble efficiently? In *COLT*, 2010. Open Problem.

S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *STOC*, pages 227–236, 2007.

N. Cesa-Bianchi and O. Shamir. Efficient online learning via randomized rounding. In *25th Annual Conference on Neural Information Processing Systems (NIPS)*, 2011.

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

S. Kakade, S. Shalev-Shwartz, and A. Tewari. Regularization techniques for learning with matrices. *JMLR*, 2012.

V. Kanade and T. Steinke. Learning hurdles for sleeping experts. In *Innovations in Theoretical Computer Science*, 2012.

R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2):245–272, 2010.

O. Shamir and S. Shalev-Shwartz. Collaborative filtering with the trace norm: Learning, bounding, and transducing. In *24th Annual Conference on Learning Theory (COLT)*, 2011.

O. Shamir and N. Srebro. Sample complexity of trace-norm? In *COLT*, 2011. Open Problem.

K. Tsuda, G. Ratsch, and M.K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6(1):995, 2006.