# Near-Optimal Algorithms for Online Matrix Prediction

Elad Hazan* Satyen Kale† Shai Shalev-Shwartz‡

**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 $(\beta, \tau)$-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 $(\beta, \tau)$-decomposable matrices. By analyzing the decomposability of cut matrices, triangular matrices, and low trace-norm matrices, we derive near optimal regret bounds for online max-cut, online gambling, and online collaborative filtering. 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 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 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

*Technion - Israel Institute of Technology. ehazan@ie.technion.ac.il.

†IBM T. J. Watson Research Center. sckale@us.ibm.com.

‡Hebrew University. shais@cs.huji.ac.il.

of matrices with a trace norm of at most $\tau$ (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 $(\beta, \tau)$-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 $(\beta, \tau)$-decomposable. Roughly speaking, $\mathbf{W}$ is $(\beta, \tau)$-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 $\tau$, and have diagonal elements bounded by $\beta$.

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, it 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 non-efficient 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 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 comparison 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 non-efficient 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.[1]

3. **Online Collaborative Filtering**: We already mentioned this problem previously. We consider the comparison class $\mathcal{W} = \{\mathbf{W} \in [-1, 1]^{m \times n} : \|\mathbf{W}\|_\star \leq \tau\}$, where $\|\cdot\|_\star$ is the trace

[1] There is no contradiction to the NP-hardness of finding the best permutation in hindsight, since we perform improper learning. That is, our algorithm is not restricted to solely using permutation matrices for prediction. The only requirement is that the regret of the algorithm relative to the class of all permutation matrices will be small.

norm. Without loss of generality assume $m \leq n$. 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 $T \geq n^2$ rounds (which means that we saw the entire matrix).

Recently, Cesa-Bianchi and Shamir [2011] proposed a rather different algorithm with regret bounded by $O(\tau\sqrt{n})$ but under the additional assumption that each entry $(i, j)$ is seen only once. In addition, while both the runtime of our method and the Cesa-Bianchi and Shamir [2011] method is 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 (see Section 6), whereas the runtime of each iteration in their algorithm is at least $\Omega(n^4)$ and can be significantly larger depending on the specific implementation.[2]

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 $(\beta, \tau)$-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 $(\beta, \tau)$-decomposability, which we define below.

**Definition 2** ($(\beta, \tau)$-decomposability)**.** *An $m \times n$ matrix $\mathbf{W}$ is $(\beta, \tau)$-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:*

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

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

[2]Specifically, each iteration in their algorithm requires solving $n$ empirical risk minimization problems over the hypothesis space of $m \times n$ matrices with a bounded trace norm (in their notation, to obtain the optimal bound, one should set $T = n^2$ and $\eta \geq 1/n$, and then should solve $\eta T$ empirical risk minimization problems per iteration). It is not clear what is the optimal runtime of solving each such empirical risk minimization problem. We believe that it is impossible to obtain a solver which is significantly faster than $n^4$.

In the above, the parameter $\beta$ stands for a **b**ound on the diagonal elements of $\mathbf{P}$ and $\mathbf{N}$, while the parameter $\tau$ stands for the **t**race of $\mathbf{P}$ and $\mathbf{N}$. It is easy to verify that if $\mathcal{W}$ is $(\beta, \tau)$-decomposable then so is its convex hull, $\mathrm{conv}(\mathcal{W})$. Throughout this paper, we assume for technical convenience that $\beta \geq 1$.[3]

There is an intriguing connection between the $(\beta, \tau)$-decomposition for a rectangular matrix $\mathbf{W}$ and its max-norm and trace norm: the least possible $\beta$ in any $(\beta, \tau)$-decomposition exactly equals half the max-norm of $\mathbf{W}$ (see Theorem 21), and the least possible $\tau$ in any $(\beta, \tau)$-decomposition exactly equals twice the trace-norm of $\mathbf{W}$ (see Theorem 23).

Our first contribution is a generic low regret algorithm for online matrix prediction with a $(\beta, \tau)$-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: β ≥ 1, τ ≥ 0, G ≥ 0
input: A set of matrices, 𝒲 ⊂ [−1, 1]^{m×n}, which is (β, τ)-decomposable
for t = 1, 2, ..., T
  adversary supplies a pair of indices (i_t, j_t) ∈ [m] × [n]
  learner picks W_t ∈ conv(𝒲) and outputs the prediction W_t(i_t, j_t)
  adversary supplies a convex, G-Lipschitz, loss function ℓ_t : [−1, 1] → ℝ
  learner pays ℓ_t(W_t(i_t, j_t))
```

**Theorem 1.** *There exists an efficient algorithm for Online Matrix Prediction which enjoys the regret bound*

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

*where $p$ is the order of $\mathsf{sym}(\mathbf{W})$ for any matrix $\mathbf{W} \in \mathcal{W}$.*

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 $y_t$.

[3]The condition $\beta \geq 1$ is not a serious restriction since for any $(\beta, \tau)$-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.

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 Appendix C) formalizes the relationship of this online problem to the max-cut problem:

**Lemma 2.** *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., 2010], 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 3.** *$\mathcal{W}$ is $(1,n)$-decomposable.*

Combining the above with Theorem 1 yields:

**Corollary 4.** *There is an efficient algorithm for the online max-cut problem with regret bounded by $2\sqrt{n \log(n) T}$.*

We prove (in Appendix 5) that the upper bound is near-optimal:

**Theorem 5.** *For any algorithm for the online max-cut problem, there is a sequence of entries $(i_t, j_t)$ and loss functions $\ell_t$ 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 $\tau$:

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

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

As before, applying the technique of Kakade et al. [2010] leads to a regret bound that scales as $\sqrt{\tau^2 T}$, which leads to trivial results in the most relevant case where $\tau = \Theta(\sqrt{mn})$. In contrast, we can obtain a much better result based on the following lemma.

**Lemma 6.** *The class $\mathcal{W}$ given in (1) is $(\sqrt{m+n}, 2\tau)$-decomposable.*

Combining the above with Theorem 1 yields:

**Corollary 7.** *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 Appendix 5) the following lower bound on the regret:

**Theorem 8.** *For any algorithm for online collaborative filtering problem with trace norm bounded by $\tau$, there is a sequence of entries $(i_t, j_t)$ and $G$-Lipschitz loss functions $\ell_t$ 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 (proved in Appendix 5).

**Theorem 9.** *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 $\mathcal{W}$.

**Lemma 10.** *The class $\mathcal{W}$ given in (2) is $(O(\log(n)), O(n\log(n)))$-decomposable.*

Combining the above with Theorem 1 yields:

**Corollary 11.** *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 12.** *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 $(\beta, \tau, \gamma)$-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:

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

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:

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

We call the above setting a $(\beta, \gamma, \tau)$-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 $(\beta, \gamma, \tau)$-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:

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

**Algorithm 1** Matrix Multiplicative Weights with Quantum Relative Entropy Projections

1: Input: $\eta$
2: Initialize $\mathbf{X}_1 = \frac{\tau}{N}\mathbf{I}$.
3: **for** $t = 1, 2, \ldots, T$: **do**
4: Play the matrix $\mathbf{X}_t$.
5: Obtain loss matrix $\mathbf{L}_t$.
6: Update $\mathbf{X}_{t+1} = \arg\min_{\mathbf{X} \in \mathcal{K}} \Delta(\mathbf{X},\ \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))$.
7: **end for**

Algorithm 1 has the following regret bound (essentially following Tsuda et al. [2006], Arora and Kale [2007], also proved in Appendix A for completeness):

**Theorem 13.** *Suppose $\eta$ is chosen so that $\eta \|\mathbf{L}_t\| \leq 1$ for all $t$ (where $\|\mathbf{L}_t\|$ is the spectral norm of $\mathbf{L}_t$). Then*

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

Equipped with the above we are ready to prove a regret bound for $(\beta, \gamma, \tau)$-OLO.

**Theorem 14.** *Assume $T \geq \frac{\tau \log(N)}{\beta}$. Then, applying Algorithm 1 with $\eta = \sqrt{\frac{\tau \log(N)}{\beta \gamma T}}$ on a $(\beta, \gamma, \tau)$-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 since the update of step 6 is a convex optimization problem. To analyze the regret of the algorithm we rely on Theorem 13. By the definition of $\mathcal{K}$ and $\mathcal{L}$, we get that $\mathbf{X}_t \bullet \mathbf{L}_t^2 \leq \beta\gamma$. Hence, the regret bound becomes

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

Substituting the value of $\eta$, we get the stated regret bound. One technical condition is that the above regret bound holds as long as $\eta$ is chosen small enough so that for all $t$, we have $\eta \|\mathbf{L}_t\| \leq 1$. Now $\|\mathbf{L}_t\| \leq \|\mathbf{L}_t\|_F = \sqrt{\text{Tr}(\mathbf{L}_t^2)} \leq \sqrt{\gamma}$. Thus, for $T \geq \frac{\tau \log(N)}{\beta}$, 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 $(\beta, \tau)$-decomposable comparison class) to a $(\beta, 4G^2, \tau)$-OCO problem with $N = 2p$. The regret bound of the derived algorithm will follow directly from Theorem 14.

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 $\mathsf{sym}(\mathbf{W})$ implies that for a pair of indices $(i, j) \in [m] \times [n]$, their corresponding indices in $\mathsf{sym}(\mathbf{W})$ are $(i, j + q)$.

Given any matrix $\mathbf{W} \in \mathcal{W}$ we embed its symmetrization $\mathsf{sym}(\mathbf{W})$ (which has size $p \times p$) into the set of $2p \times 2p$ positive semidefinite matrices as follows. Since $\mathbf{W}$ admits a $(\beta, \tau)$-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[4]

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

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

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

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 Appendix 6 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 $\ell_t$. Given the loss function $\ell_t$, let $g$ be a subderivative of $\ell_t$ at $\hat{y}_t$. Since $\ell_t$ 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 $(i_t + q, i_t + q)$, $(j_t + q, j_t + q)$, $(p + i_t + q, p + i_t + q)$, and $(p + j_t + q, p + j_t + q)$, all equalling $g^2$. Hence, $\mathrm{Tr}(\mathbf{L}_t^2) = 4g^2 \leq 4G^2$.

[4] 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 $\mathbf{W}$.

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

**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 (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 $\ell_t$ at $\hat{y}_t$
    Let $\mathbf{L}_t$ be as defined in (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

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

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

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

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

### 3.2.1 Proof of Theorem 1

Following our reduction, we can now appeal to Theorem 14. For $T \geq \frac{\tau \log(2p)}{\beta}$, the bound of Theorem 14 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, collaborative filtering, and gambling. All the three decompositions we give are optimal up to constant factors.

## 4.1 Proof of Lemma 3 (max-cut)

We need to show that every matrix $\mathbf{W}_A \in \mathcal{W}$ admits a $(1, n)$-decomposition. We can rewrite $\mathbf{W}_A = -\mathbf{w}_A\mathbf{w}^\top$ 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 6 (collaborative filtering)

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 Lemma 19).

**Theorem 15.** *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

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

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 Theorem 20 (proved in Appendix D), 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 15 is optimal up to a constant factor.

### 4.3 Proof of Lemma 10 (gambling)

We need to show that every matrix $\mathbf{W} \in \mathcal{W}$, i.e. an $n \times n$ matrix $\mathbf{W}_\pi$ for some permutation $\pi : [n] \to [n]$, admits a $(O(\log(n)), O(n\log(n)))$-decomposition. One minor change that needs to be made to Algorithm 2 is that the last constraint in (3) needs to be changed to

$$\forall (i,j) \in [n] \times [n] : \ (X(i, j+q) - X(p+i, p+j+q)) \ \in [0,1],$$

to ensure that the prediction lies in $[0,1]$ rather than $[-1,1]$. The analysis remains intact, and so does the regret bound.

We now give the decomposition. The following upper triangular matrix $\mathbf{T}$ plays a pivotal role:

$$T(i,j) = \begin{cases} 1 & \text{if } i \leq j \\ 0 & \text{otherwise.} \end{cases}$$

The reason this matrix is so important is because any matrix $\mathbf{W}_\pi$ is obtained by permuting the rows and columns of $\mathbf{T}$. In particular, let $\mathbf{P}_\pi$ be the permutation matrix defined by the permutation $\pi$, i.e.

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

Then it is easy to check that

$$\mathbf{W}_\pi = \mathbf{P}_\pi \mathbf{T} \mathbf{P}_\pi^\top.$$

Using this fact, we get

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathbf{P}_\pi & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\pi \end{bmatrix}}_{\mathbf{Q}_\pi} \mathsf{sym}(\mathbf{T}) \begin{bmatrix} \mathbf{P}_\pi^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\pi^\top \end{bmatrix} &= \begin{bmatrix} \mathbf{P}_\pi & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\pi \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_\pi^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\pi^\top \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{P}_\pi \mathbf{T} \mathbf{P}_\pi^\top \\ \mathbf{P}_\pi \mathbf{T}^\top \mathbf{P}_\pi^\top & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{W}_\pi \\ \mathbf{W}_\pi^\top & \mathbf{0} \end{bmatrix} = \mathsf{sym}(\mathbf{W}_\pi). \end{aligned}$$

Now, note that $\mathbf{Q}_\pi$ is a permutation matrix (viz. the one defined by the permutation $\pi' : [2n] \to [2n]$ defined as $\pi'(i) = \pi(i)$ for $1 \leq i \leq n$, and $\pi'(i) = \pi(i-n) + n$ for $n < i \leq 2n$). Thus, if $\mathbf{T}$ admits a $(\beta, \tau)$-decomposition, $\mathsf{sym}(\mathbf{T}) = \mathbf{P} - \mathbf{N}$, then

$$\mathsf{sym}(\mathbf{W}_\pi) \ = \ \mathbf{Q}_\pi \mathsf{sym}(\mathbf{T}) \mathbf{Q}_\pi^\top \ = \ \mathbf{Q}_\pi \mathbf{P} \mathbf{Q}_\pi^\top - \mathbf{Q}_\pi \mathbf{N} \mathbf{Q}_\pi^\top$$

is a $(\beta, \tau)$-decomposition for $\mathsf{sym}(\mathbf{W}_\pi)$. This is because the diagonal entries of $\mathbf{Q}_\pi \mathbf{P} \mathbf{Q}_\pi^\top$ (resp. $\mathbf{Q}_\pi \mathbf{N} \mathbf{Q}_\pi^\top$) are simply a permutation (viz. $\pi'$) of the diagonal entries of $\mathbf{P}$ (resp. $\mathbf{N}$). Since

$\mathbf{ABA}^\top \succeq \mathbf{0}$ if $\mathbf{B} \succeq \mathbf{0}$ for any matrix $\mathbf{A}$, the matrices $\mathbf{Q}_\pi \mathbf{P} \mathbf{Q}_\pi^\top$ and $\mathbf{Q}_\pi \mathbf{P} \mathbf{Q}_\pi^\top$ are both positive semidefinite.

So now we show that $\mathbf{T}$ admits a $(O(\log(n)), O(n \log(n)))$-decomposition. For convenience, we assume that $n$ is a power of 2, i.e. $n = 2^k$ for some integer $k \geq 0$. For $n$ that are not a power of 2, we can readily obtain a decomposition by the following observation: if we take the smallest power of 2 that is larger than $n$, say $2^k$, and consider the symmetrized triangular matrix for $2^k$, then $\mathsf{sym}(\mathbf{T})$ can be expressed as a principal submatrix of it. Then taking the corresponding principal submatrices from the decomposition for the triangular matrix for $2^k$ we obtain a decomposition for $n$. This uses the fact that principal submatrices of positive semidefinite matrices are positive semidefinite as well.

**Theorem 16.** *Let $n = 2^k$ for some integer $k \geq 0$. Then $\mathbf{T}$ admits a $(k+1, 4n(k+1))$-decomposition.*

*Proof.* We show that $\mathsf{sym}(\mathbf{T})$ can be written as a difference of positive semidefinite matrices with diagonals bounded by $k+1$. The bound on the sum of traces, $4n(k+1)$, of the two matrices follows trivially.

We use a recursive construction. Let the triangular matrix for $n = 2^k$ be denoted by $\mathbf{T}_k$. For $k = 0$, the following is a decomposition for $\mathbf{T}_0$ with diagonals bounded by 1:

$$\mathsf{sym}(\mathbf{T}_0) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

So now assume that $k > 0$ and we have a decomposition for $\mathbf{T}_{k-1}$ with diagonals bounded by $k$, i.e.

$$\mathsf{sym}(\mathbf{T}_{k-1}) \;=\; \begin{bmatrix} \mathbf{0} & \mathbf{T}_{k-1} \\ \mathbf{T}_{k-1}^\top & \mathbf{0} \end{bmatrix} \;=\; \mathbf{P} - \mathbf{N},$$

where $\mathbf{P}, \mathbf{N} \succeq \mathbf{0}$, and for all $i \in [2^k]$, $P(i,i), N(i,i) \leq k$. We need the following block decomposition of $\mathbf{P}$ and $\mathbf{N}$ into contiguous $2^{k-1} \times 2^{k-1}$ blocks as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^A & \mathbf{P}^B \\ \mathbf{P}^C & \mathbf{P}^D \end{bmatrix} \text{ and } \mathbf{N} = \begin{bmatrix} \mathbf{N}^A & \mathbf{N}^B \\ \mathbf{N}^C & \mathbf{N}^D \end{bmatrix}.$$

Then we have the following decomposition of $\mathsf{sym}(\mathbf{T}_k)$. All the blocks in the decomposition below are of size $2^{k-1} \times 2^{k-1}$.

$$\mathsf{sym}(\mathbf{T}_k) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{T}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_{k-1} \\ \mathbf{T}_{k-1}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{k-1}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Now, consider the following decompositions of the two matrices above as a difference of positive semidefinite matrices. For the first matrix, the diagonals in the decomposition are bounded by 1:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}.$$

For the second matrix, the diagonals in the decomposition are bounded by $k$.

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{T}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_{k-1} \\ \mathbf{T}_{k-1}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{k-1}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B \\ \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D \end{bmatrix} - \begin{bmatrix} \mathbf{N}^A & \mathbf{0} & \mathbf{N}^B & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^A & \mathbf{0} & \mathbf{N}^B \\ \mathbf{N}^C & \mathbf{0} & \mathbf{N}^D & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^C & \mathbf{0} & \mathbf{N}^D \end{bmatrix}.$$

It is easy to verify that the matrices in the decomposition above are positive semidefinite, since each is a sum of two positive semidefinite matrices. For example:

$$\begin{bmatrix} \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B \\ \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D \end{bmatrix} = \begin{bmatrix} \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^A & \mathbf{0} & \mathbf{P}^B \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^C & \mathbf{0} & \mathbf{P}^D \end{bmatrix}.$$

Adding the two decompositions, we get a decomposition for $\mathsf{sym}(\mathbf{T}_k)$ as a difference of two positive semidefinite matrices. The diagonal entries of these two matrices are bounded by $k+1$, as required. □

This decomposition is optimal up to constant factors. This is because the singular values of $\mathbf{T}$ are $\frac{1}{2\cos(\frac{k\pi}{2n+1})}$ for $k = 1, 2, \ldots, n$ (see Elkies [2011]). This implies that $\|\mathbf{T}\|_\star = \Theta(n \log(n))$. Thus, the best $\beta$ one can get is $\Theta(\log(n))$, and the best $\tau$ is $\Theta(n \log(n))$.

# 5 Lower bounds

In this section we prove the lower bounds stated in Section 2.

## 5.1 Online Max Cut

We prove Theorem 5, which we restate here for convenience:

> **Theorem 5 restated:** For any algorithm for the online max cut problem, there is a sequence of entries $(i_t, j_t)$ and loss functions $\ell_t$ for $t = 1, 2, \ldots, T$ such that the regret of the algorithm is at least $\sqrt{nT/16}$.

*Proof.* Consider the following stochastic adversary. Divide up the time period $T$ into $n/2$ equal size[5] intervals $T_i$, for $i \in [n/2]$, corresponding to the $n/2$ pairs of indices $(i, i+n/2)$ for $i \in [n/2]$. For every $i \in [n/2]$ and for each $t \in T_i$, the adversary sets $(i_t, j_t) = (i, i+n/2)$ and $y_t$ to be a Rademacher random variable independent of all other such variables. Clearly, the expected regret of any algorithm for the online max cut problem equals $\frac{T}{2}$.

Now, define the following subset of vertices $A$: for every $i \in [n/2]$, consider $S_i = \sum_{t \in T_i} y_t$. If $S_i < 0$, include both $i, i+n/2 \in A$, else only include $i \in A$. By construction, the matrix $\mathbf{W}_A$ has the following property for all $i \in [n/2]$:

$$W_A(i, i+n/2) \;=\; \mathrm{sgn}(S_i).$$

[5] We assume for convenience that $\frac{n}{2}$ and $\frac{2T}{n}$ are integers.

Using the definition of $\ell_t$ and the fact that $|T_i| = 2T/n$, we obtain

$$\begin{aligned}\mathbf{E}\left[\sum_{t\in T_i}\ell_t(W_A(i,i+n/2))\right] &= \mathbf{E}\left[\sum_{t\in T_i}\left(\tfrac{1}{2}-\tfrac{\mathrm{sgn}(S_i)}{2}y_t\right)\right]\\ &= \mathbf{E}\left[\frac{T}{n}-\frac{|S_i|}{2}\right] \le \frac{T}{n}-\sqrt{\frac{T}{4n}},\end{aligned}$$

where we used Khintchine's inequality: if $X$ is a sum of $k$ independent Rademacher random variables, then $\mathbf{E}[|X|] \ge \sqrt{k/2}$. Summing up over all $i \in [n/2]$, we get that

$$\mathbf{E}\left[\sum_{t=1}^{T}\ell_t(W_A(i_t,j_t))\right] \le \frac{n}{2}\left[\frac{T}{n}-\sqrt{\frac{T}{4n}}\right] = \frac{T}{2}-\sqrt{\frac{nT}{16}}.$$

Hence the expected regret of the algorithm is at least $\sqrt{\frac{nT}{16}}$. In particular, there is a setting of the $\hat{y}_t$ variables so that the regret of the algorithm is at least $\sqrt{\frac{nT}{16}}$. □

### 5.2 Online Collaborative Filtering with Bounded Trace Norm

We start with the proof of Theorem 8, which we restate here for convenience:

> **Theorem 8 restated:** For any algorithm for online collaborative filtering problem with trace norm bounded by $\tau$, there is a sequence of entries $(i_t, j_t)$ and loss functions $\ell_t$ for $t = 1, 2, \ldots, T$ such that the regret of the algorithm is at least $G\sqrt{\frac{1}{2}\tau\sqrt{n}T}$.

*Proof.* First, we may assume that $\tau \le m\sqrt{n}$: this is because for any matrix $\mathbf{W} \in [-1,1]^{m\times n}$, we have

$$\|\mathbf{W}\|_\star \le \sqrt{\mathrm{rank}(\mathbf{W})}\|\mathbf{W}\|_F \le \sqrt{m}\cdot\sqrt{mn} = m\sqrt{n},$$

since $\mathrm{rank}(\mathbf{W}) \le m$. So now we focus on the sub-matrix formed by the first $\frac{\tau}{\sqrt{n}}$ rows[6] and all $n$ columns. This sub-matrix has $\tau\sqrt{n}$ entries.

Consider the following stochastic adversary. Divide up the time period $T$ into $\tau\sqrt{n}$ intervals of length $\frac{T}{\tau\sqrt{n}}$, indexed by $\tau\sqrt{n}$ pairs $(i,j)$ corresponding to the entries of the sub-matrix. For every $(i,j)$, and for every round $t$ in the interval $I_{ij}$ corresponding to $(i,j)$, we set the loss function to be $\ell_t(\mathbf{W}) = \sigma_t G W_{ij}$, where $\sigma_t \in \{-1,1\}$ is a Rademacher random variable chosen independently of all other such variables. Note that the absolute value of derivative of the loss function is $G$.

Clearly, any algorithm for OCF has expected loss 0. Now consider the matrix $\mathbf{W}^\star$ where

$$\forall i \in \left[\frac{\tau}{\sqrt{n}}\right], j \in [n]: \; W^\star_{ij} = -\mathrm{sgn}\left(\textstyle\sum_{t\in I_{ij}}\sigma_t\right),$$

and all entries in rows $i > \frac{\tau}{\sqrt{n}}$ are set to 0. Since $\mathrm{rank}(\mathbf{W}^\star) \le \frac{\tau}{\sqrt{n}}$, we have

$$\|\mathbf{W}^\star\|_\star \le \sqrt{\mathrm{rank}(\mathbf{W}^\star)}\cdot\|\mathbf{W}^\star\|_F \le \sqrt{\frac{\tau}{\sqrt{n}}}\cdot\sqrt{\tau\sqrt{n}} = \tau,$$

[6] For convenience, we assume that $\frac{\tau}{\sqrt{n}}$ and $\frac{T}{\tau\sqrt{n}}$ are integers.

so $\mathbf{W}^\star \in \mathcal{W}$.[7]

The expected loss of $\mathbf{W}^\star$ is

$$\begin{aligned}
\sum_{ij} \mathbb{E}\left[\sum_{t\in I_{ij}} \sigma_t G W^\star_{ij}\right] &= G\sum_{ij}\mathbb{E}\left[-\left|\sum_{t\in I_{ij}}\sigma_t\right|\right] \\
&\geq -G\sum_{ij}\sqrt{\frac{1}{2}|I_{ij}|} \\
&= -G\tau\sqrt{n}\cdot\sqrt{\frac{T}{2\tau\sqrt{n}}} \\
&= -G\sqrt{\frac{1}{2}\tau\sqrt{n}T},
\end{aligned}$$

where the inequality above is again due to Khintchine's inequality. Hence, the expected regret of the algorithm is at least $G\sqrt{\frac{1}{2}\tau\sqrt{n}T}$. In particular, there is a specific assignment of values to $\sigma_t$ such that the regret of the algorithm is at least $G\sqrt{\frac{1}{2}\tau\sqrt{n}T}$. □

The construction we used for deriving the above lower bound can be easily adapted to derive a lower bound on the sample complexity of learning the class $\mathcal{W}$ in the batch setting. This is formalized in Theorem 9, which we restate here for convenience.

> **Theorem 9 restated** 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 $\Omega(n^{1.5}/\varepsilon^2)$.

*Proof.* For simplicity, let us choose $m = n$. Let $k = \tau/\sqrt{n}$ and fix some small $\varepsilon$. Define a family of distributions over $[n]^2 \times \{-1, 1\}$ as follows. Each distribution is parameterized by a matrix $\mathbf{W}$ such that there is some $I \subset [n]$, with $|I| = k$, where $W(i,j) \in \{-1,1\}$ for $i \in I$ and $W(i,j) = 0$ for $i \notin I$. Now, the probability to sample an example $(i,j,y)$ is $\left(\frac{1}{2} + 2\varepsilon\right)\frac{1}{kn}$ if $i \in I$ and $y = W(i,j)$, is $\left(\frac{1}{2} - 2\varepsilon\right)\frac{1}{kn}$ if $i \in I$ and $y = -W(i,j)$, and the probability is 0 in all other cases.

As in the proof of Theorem 8, any matrix defining such distribution is in $\mathcal{W}$. Furthermore, if we consider the absolute loss function: $\ell(\mathbf{W}, (i,j,y)) = \frac{1}{2}|W(i,j) - y|$, then the expected loss of $\mathbf{W}$ with respect to the distribution it defines is

$$\mathbf{E}\left[\tfrac{1}{2}|W(i,j) - y|\right] = \tfrac{1}{2} - 2\varepsilon\ .$$

In contrast, by standard no-free-lunch arguments, no algorithm can know to predict an entry $(i,j)$ with error smaller than $\frac{1}{2} - \varepsilon$ without observing $\Omega(1/\varepsilon^2)$ examples from this entry. Therefore, no algorithm can have an error smaller than $\frac{1}{2} - \varepsilon$ without receiving $\Omega(kn/\varepsilon^2)$ examples. □

# 6 Implementation Details

In general, the update rule in Algorithm 1 is a convex optimization problem and can be computed in polynomial time. We now give the following more efficient implementation which takes essentially

[7] This construction is tight: e.g. if $\mathbf{W}^\star$ is formed by taking $\frac{t}{\sqrt{n}}$ rows of an $n \times n$ Hadamard matrix.

$\tilde{O}(p^3)$ time per round. This is based on the following theorem that is essentially proved in Tsuda et al. [2006]:

**Theorem 17.** *The optimal solution of* $\arg\min_{\mathbf{X}\in\mathcal{K}} \Delta(\mathbf{X}, \mathbf{Y})$*, where* $\mathbf{Y}$ *is a given symmetric matrix, and*

$$\mathcal{K} \ := \ \{\mathbf{X} \in \mathbb{S}^{n\times n} : \ \mathbf{A}_j \bullet \mathbf{X} \leq b_j \text{ for } j = 1, 2, \ldots, m\},$$

*is given by*

$$\mathbf{X}^\star = \exp(\log(\mathbf{Y}) - \textstyle\sum_{j=1}^m \alpha_j^\star \mathbf{A}'_j),$$

*where* $\mathbf{A}'_j = \frac{1}{2}(\mathbf{A}_j + \mathbf{A}_j^\top)$*, and* $\boldsymbol{\alpha}^\star = \langle \alpha_1^\star, \alpha_2^\star, \ldots, \alpha_m^\star \rangle$ *is given by*

$$\boldsymbol{\alpha}^\star \ = \ \arg \max_{\forall j\in[m]:\ \alpha_j \geq 0} -\mathrm{Tr}(\exp(\log(\mathbf{Y}) - \textstyle\sum_{j=1}^m \alpha_j \mathbf{A}'_j)) - \sum_{j=1}^m \alpha_j b_j.$$

The idea is to avoid taking projections on the set $\mathcal{K}$ in each round. If the chosen entry in round $t$ is $(i_t, j_t)$, then we compute $\mathbf{X}_t$ as

$$\mathbf{X}_t \ = \ \arg\min_{\mathbf{X}\in\mathcal{K}_t} \Delta(\mathbf{X}, \exp(\log(\mathbf{X}_{t-1} - \eta \mathbf{L}_{t-1})),$$

where the polytope $\mathcal{K}_t$ is defined as

$$\begin{aligned}
\mathcal{K}_t \ := \ \Bigg\{ & \ \mathbf{X} \in \mathbb{S}^{2p\times 2p} \ \text{ s.t.} \\
& X(i_t, i_t) + X(j_t + q, j_t + q) + X(p + i_t, p + i_t) + X(p + j_t + q, p + j_t + q) \leq 4\beta \\
& X(i_t, j_t + q) - X(p + i_t, p + j_t + q)) \ \leq \ 1 \\
& X(p + i_t, p + j_t + q)) - X(i_t, j_t + q) \ \leq \ 1 \\
& \mathrm{Tr}(\mathbf{X}) \ \leq \ \tau \ \Bigg\}
\end{aligned}$$

The observation is that this suffices for the regret bound of Theorem 14 to hold since the optimal point in hindsight $\mathbf{X}^\star \in \mathcal{K}_t$ for all $t$ (see the proof of Theorem 13).

Note that $\mathcal{K}_t$ is defined using just 4 constraints, and hence the dual problem given in Theorem 17 has only 4 variables $\alpha_j$. Thus, standard convex optimization techniques (say, the ellipsoid method) can be used to solve the dual problem to $\varepsilon$-precision in $O(\log(1/\varepsilon))$ iterations, each of which requires computing the gradient and/or the Hessian of the objective, which can be done in $O(p^3)$ time via the eigendecomposition, leading to an $\tilde{O}(p^3)$ time algorithm overall.

More precisely, the iteration count for convex optimization methods have logarithmic dependence on the range of the $\alpha_j$ variables. Since $\mathrm{Tr}(\mathbf{X}_{t-1}) \leq \tau$, we see (using the Golden-Thompson inequality [Golden, 1965, Thompson, 1965]) that

$$\mathrm{Tr}(\exp(\log(\mathbf{X}_{t-1} - \eta \mathbf{L}_{t-1}))) \ \leq \ \mathbf{X}_{t-1} \bullet \exp(-\eta \mathbf{L}_{t-1}) \ \leq \ 3\tau.$$

Thus, setting all $\alpha_j = 0$, the dual objective value is at least $-3\tau$. Since $b_j \geq 1$ for all $j$, we get that the optimal values of $\alpha_j$ are all bounded by $3\tau$. Thus, the range of all $\alpha_j$ can be set to $[0, 3\tau]$, giving a $O(\log(\frac{\tau}{\varepsilon}))$ bound on the number of iterations.

# 7 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 $(\beta, \tau)$-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.

N. D. Elkies. 2-norm of the upper triangular "all-ones" matrix. http://mathoverflow.net/questions/72361/2-norm-of-the-upper-triangular-all-ones-matrix, 2011.

S. Golden. Lower Bounds for the Helmholtz Function. *Physical Review*, 137:1127–1128, February 1965. doi: 10.1103/PhysRev.137.B1127.

S. Kakade, S. Shalev-Shwartz, and A. Tewari. Regularization techniques for learning with matrices. *preprint arXiv:0910.0610*, 2010.

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.

J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. A. Tropp. Practical large-scale optimization for max-norm regularization. In *NIPS*, pages 1297–1305, 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.

C. J. Thompson. Inequality with applications in statistical mechanics. *Journal of Mathematical Physics*, 6(11):1812–1823, 1965.

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.

# A Matrix Multiplicative Weights Algorithm

For the sake of completeness, we prove Theorem 13. The setting is as follows. We have an online convex optimization problem where the decision set is a convex subset $\mathcal{K}$ of $N \times N$ positive semidefinite matrices of trace bounded by $\tau$, viz. for all $\mathbf{X} \in \mathcal{K}$, we have $\mathbf{X} \succeq \mathbf{0}$ and $\mathrm{Tr}(\mathbf{X}) \leq \tau$. We assume for convenience that $\frac{\tau}{N}\mathbf{I} \in \mathcal{K}$. In each round $t$, the learner produces a matrix $\mathbf{X}_t \in \mathcal{K}$, and the adversary supplies a loss matrix $\mathbf{L}_t \in \mathbb{R}^{N \times N}$, which is assumed to be symmetric. The loss of the learner is $\mathbf{X}_t \bullet \mathbf{L}_t$. The goal is to minimize regret defined as

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

Consider Algorithm 1. We now prove Theorem 13, which we restate here for convenience:

**Theorem 18.** *Suppose $\eta$ is chosen so that $\eta \|\mathbf{L}_t\| \leq 1$ for all $t$. Then*

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

*Proof.* Consider any round $t$. Let $\mathbf{X} \in \mathcal{K}$ be any matrix. We use the quantum relative entropy, $\Delta(\mathbf{X}, \mathbf{X}_t)$, as a potential function. We have

$$\Delta(\mathbf{X}, \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t)) - \Delta(\mathbf{X}, \mathbf{X}_t) \ = \ \eta \mathbf{X} \bullet \mathbf{L}_t - \mathrm{Tr}(\mathbf{X}_t) + \mathrm{Tr}(\exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t)). \tag{5}$$

Now quantum relative entropy projection onto the set $\mathcal{K}$ is a Bregman projection, and hence the Generalized Pythagorean inequality applies (see Tsuda et al. [2006]):

$$\Delta(\mathbf{X}, \mathbf{X}_{t+1}) + \Delta(\mathbf{X}_{t+1}, \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))) \ \leq \ \Delta(\mathbf{X}, \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))),$$

and since $\Delta(\mathbf{X}_{t+1}, \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))) \geq 0$, we get that

$$\Delta(\mathbf{X}, \mathbf{X}_{t+1}) \ \leq \ \Delta(\mathbf{X}, \exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t))).$$

Hence from (5) we get

$$\Delta(\mathbf{X}, \mathbf{X}_{t+1}) - \Delta(\mathbf{X}, \mathbf{X}_t) \;\leq\; \eta \mathbf{X} \bullet \mathbf{L}_t - \mathrm{Tr}(\mathbf{X}_t) + \mathrm{Tr}(\exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t)). \tag{6}$$

Now, using the Golden-Thompson inequality [Golden, 1965, Thompson, 1965], we have

$$\mathrm{Tr}(\exp(\log(\mathbf{X}_t) - \eta \mathbf{L}_t)) \;\leq\; \mathrm{Tr}(\mathbf{X}_t \exp(-\eta \mathbf{L}_t))$$

Next, using the fact that $\exp(\mathbf{A}) \preceq \mathbf{I} + \mathbf{A} + \mathbf{A}^2$ for $\|\mathbf{A}\| \leq 1$,[8] we obtain

$$\begin{aligned} \mathrm{Tr}(\mathbf{X}_t \exp(-\eta \mathbf{L}_t)) \leq\; & \mathrm{Tr}(X_t(\mathbf{I} - \eta \mathbf{L}_t + \eta^2 \mathbf{L}_t^2) \\ =\; & \mathrm{Tr}(\mathbf{X}_t) - \eta \mathbf{X}_t \bullet \mathbf{L}_t + \eta^2 \mathbf{X}_t \bullet \mathbf{L}_t^2. \end{aligned}$$

Combining the above and plugging into (6) we get

$$\Delta(\mathbf{X}, \mathbf{X}_{t+1}) - \Delta(\mathbf{X}, \mathbf{X}_t) \;\leq\; \eta \mathbf{X} \bullet \mathbf{L}_t - \eta \mathbf{X}_t \bullet \mathbf{L}_t + \eta^2 \mathbf{X}_t \bullet \mathbf{L}_t^2. \tag{7}$$

Summing up from $t = 1$ to $T$, and rearranging, we get

$$\begin{aligned} \text{Regret} \;\leq\; & \eta \sum_{t=1}^{T} \mathbf{X}_t \bullet \mathbf{L}_t^2 + \frac{\Delta(\mathbf{X}, \mathbf{X}_1) - \Delta(\mathbf{X}, \mathbf{X}_{T+1})}{\eta} \\ \leq\; & \eta \sum_{t=1}^{T} \mathbf{X}_t \bullet \mathbf{L}_t^2 + \frac{\tau \log(N)}{\eta}, \end{aligned}$$

since $\Delta(\mathbf{X}, \mathbf{X}_{T+1}) \geq 0$ and

$$\begin{aligned} \Delta(\mathbf{X}, \mathbf{X}_1) \;=\; & \mathbf{X} \bullet (\log(\mathbf{X}) - \log(\tfrac{\tau}{N}\mathbf{I})) - \mathrm{Tr}(\mathbf{X}) + \tau \\ =\; & \mathbf{X} \bullet \log(\tfrac{1}{\tau}\mathbf{X}) + \log(\tau)\mathrm{Tr}(\mathbf{X}) - \log(\tfrac{\tau}{N})\mathrm{Tr}(\mathbf{X}) - \mathrm{Tr}(\mathbf{X}) + \tau \\ \leq\; & \mathrm{Tr}(\mathbf{X})(\log(N) - 1) + \tau \\ \leq\; & \tau \log(N). \end{aligned}$$

The first inequality above follows because $\mathrm{Tr}(\mathbf{X}) \leq \tau$, so $\log(\frac{1}{\tau}\mathbf{X}) \prec \mathbf{0}$. The second inequality uses $\mathrm{Tr}(\mathbf{X}) \leq \tau$.

□

# B Technical Lemmas and Proofs

**Lemma 19.** *For $m \times n$ non-symmetric matrices $\mathbf{W}$, if $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the singular value decomposition of $\mathbf{W}$, then*

$$\mathsf{sym}(\mathbf{W}) = \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U} & \frac{1}{\sqrt{2}}\mathbf{U} \\ \frac{1}{\sqrt{2}}\mathbf{V} & -\frac{1}{\sqrt{2}}\mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & -\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U}^\top & \frac{1}{\sqrt{2}}\mathbf{V}^\top \\ \frac{1}{\sqrt{2}}\mathbf{U}^\top & -\frac{1}{\sqrt{2}}\mathbf{V}^\top \end{bmatrix}$$

*is the eigenvalue decomposition of $\mathsf{sym}(\mathbf{W})$. In particular, $\|\mathsf{sym}(\mathbf{W})\|_\star = 2\|\mathbf{W}\|_\star$.*

[8] To see this, note that we can write $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$ for some orthonormal $\mathbf{V}$ and diagonal $\mathbf{D}$. Therefore,

$$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 - e^{\mathbf{A}} = \mathbf{V}\left(\mathbf{I} + \mathbf{D} + \mathbf{D}^2 - e^{\mathbf{D}}\right)V^\top .$$

Now, by the inequality $1 + a + a^2 - e^a \geq 0$, which holds for all $a \leq 1$, we obtain that all elements of the diagonal matrix $\left(\mathbf{I} + \mathbf{D} + \mathbf{D}^2 - e^{\mathbf{D}}\right)$ are non-negative.

*Proof.* By the block matrix multiplication rule we have

$$\begin{aligned}
&\begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U} & \frac{1}{\sqrt{2}}\mathbf{U} \\ \frac{1}{\sqrt{2}}\mathbf{V} & -\frac{1}{\sqrt{2}}\mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & -\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U}^\top & \frac{1}{\sqrt{2}}\mathbf{V}^\top \\ \frac{1}{\sqrt{2}}\mathbf{U}^\top & -\frac{1}{\sqrt{2}}\mathbf{V}^\top \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U}\mathbf{\Sigma} & -\frac{1}{\sqrt{2}}\mathbf{U}\mathbf{\Sigma} \\ \frac{1}{\sqrt{2}}\mathbf{V}\mathbf{\Sigma} & \frac{1}{\sqrt{2}}\mathbf{V}\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U}^\top & \frac{1}{\sqrt{2}}\mathbf{V}^\top \\ \frac{1}{\sqrt{2}}\mathbf{U}^\top & -\frac{1}{\sqrt{2}}\mathbf{V}^\top \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} & \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\ \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top & \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{0} \end{bmatrix}.
\end{aligned}$$

In addition, it is easy to check that the columns of $\begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{U} & \frac{1}{\sqrt{2}}\mathbf{U} \\ \frac{1}{\sqrt{2}}\mathbf{V} & -\frac{1}{\sqrt{2}}\mathbf{V} \end{bmatrix}$ are orthonormal. It follows that the above form is the eigendecomposition of $\mathsf{sym}(\mathbf{W})$. Therefore, for any Schatten norm: $\|\mathsf{sym}(\mathbf{W})\| = 2\|\mathbf{\Sigma}\| = 2\|\mathbf{W}\|$, which concludes our proof. □

# C The optimal cut in the Online Max Cut problem

We prove Lemma 2, which we restate here for convenience.

**Lemma 2 restated** Consider an online sequence of loss functions $\{\ell_t = \frac{1}{2}|y_t - \hat{y}_y|\}$. 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 $(i, j)$.

*Proof.* Consider $\mathbf{W}_A$. For each pair $(i, j)$ let $c^+_{ij}, c^-_{ij}$ be the total number of iterations in which the pair $(i, j)$ appeared in the adversarial sequence with $y_t = 1$ or $y_t = -1$ respectively. Since $\hat{y}_t \in [-1, 1]$ we can rewrite the total loss as:

$$\begin{aligned}
\sum_t \ell_t(\mathbf{W}_A(i_t, j_t)) &= \frac{1}{2}\sum_{(i,j)} [c^+_{ij} \cdot (1 - W_A(i, j)) + c^-_{ij} \cdot (1 + W_A(i, j))] \\
&= \frac{1}{2}\sum_{(i,j)} W_A(i, j) \cdot (c^-_{ij} - c^+_{ij}) + C_T \\
&= -\frac{1}{2}\sum_{(i,j)} W_A(i, j) \cdot w_{ij} + C_T
\end{aligned}$$

Where $C_T$ is a constant which is independent of $\mathbf{W}_A$. Hence, minimizing the above expression is equivalent to maximizing the expression:

$$\sum_{(i,j)} W_A(i, j) \cdot w_{ij} = 2 \cdot \sum_{(i,j):\ W_A(i,j)=1} w_{ij} - \sum_{(i,j)} w_{ij}.$$

Since $\sum_{(i,j)} w_{ij}$ is a constant independent of $A$, the cut which maximizes this expression is the maximum cut in the weighted graph over the weights $w_{ij}$. □

# D Optimality of Decomposition for Collaborative Filtering

In this section, we prove the following theorem:

**Theorem 20.** *Consider the matrix* $\mathbf{W}$ *formed by taking* $m = \frac{\tau}{\sqrt{n}}$ *rows of an* $n \times n$ *Hadamard matrix. This matrix has* $\|\mathbf{W}\|_\star = \tau$*, and any* $(\beta, \tilde{\tau})$*-decomposition for* $\mathsf{sym}(\mathbf{W})$ *has*

$$\beta\tilde{\tau} \;\geq\; \frac{1}{4}\tau\sqrt{n}.$$

*Proof.* Since the rows of $\mathbf{W}$ are orthogonal to each other, the $m$ singular values of $\mathbf{W}$ all equal $\sqrt{n}$, and thus $\|\mathbf{W}\|_\star = m\sqrt{n} = \tau$. Further, the SVD of $\mathbf{W}$ is (here, $\mathbf{I}_m$ is the $m \times m$ identity matrix):

$$\mathbf{W} \;=\; \mathbf{I}_m(\sqrt{n}\mathbf{I}_m)(\tfrac{1}{\sqrt{n}}\mathbf{W}).$$

Using Lemma 19 the eigendecomposition of $\mathsf{sym}(\mathbf{W})$ can be written as

$$\mathsf{sym}(\mathbf{W}) \;=\; \mathbf{U}(\sqrt{n}\mathbf{I}_m)\mathbf{U}^\top + \mathbf{V}(-\sqrt{n}\mathbf{I}_m)\mathbf{V}^\top,$$

where

$$\mathbf{U} = [\tfrac{1}{\sqrt{2}}\mathbf{I}_m, \tfrac{1}{\sqrt{2n}}\mathbf{W}]^\top \text{ and } \mathbf{V} = [\tfrac{1}{\sqrt{2}}\mathbf{I}_m, -\tfrac{1}{\sqrt{2n}}\mathbf{W}]^\top$$

are $p \times m$ matrices with orthonormal columns.

Let $\mathsf{sym}(\mathbf{W}) = \mathbf{P} - \mathbf{N}$ be a $(\beta, \tilde{\tau})$-decomposition. Now consider the following matrices: first, define the $p \times p$ diagonal matrix

$$\mathbf{D} \;:=\; \begin{bmatrix} \frac{1}{\sqrt{2m}}\mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \frac{\sqrt{mn}}{2\sqrt{2}\tilde{\tau}}\mathbf{I}_n \end{bmatrix}.$$

Finally, define the $p \times p$ positive semidefinite matrix

$$\mathbf{Y} \;:=\; \mathbf{D}\mathbf{U}\mathbf{U}^\top\mathbf{D}.$$

Since $\mathbf{U}$ has orthonormal columns we have $\mathbf{U}\mathbf{U}^\top \preceq \mathbf{I}_p$, and so

$$\mathbf{Y} \;\preceq\; \mathbf{D}\mathbf{I}_p\mathbf{D} \;=\; \mathbf{D}^2.$$

Now, consider

$$\begin{aligned}
\mathbf{Y} \bullet \mathsf{sym}(\mathbf{W}) \;&=\; \mathbf{Y} \bullet (\mathbf{P} - \mathbf{N}) \\
&\leq\; \mathbf{Y} \bullet \mathbf{P} && (\because \mathbf{Y}, \mathbf{N} \succeq \mathbf{0},\ \text{so } \mathbf{Y} \bullet \mathbf{N} \geq 0) \\
&\leq\; \mathbf{D}^2 \bullet \mathbf{P} && (\because \mathbf{Y} \preceq \mathbf{D}^2) \\
&=\; \sum_{i=1}^{m} \frac{1}{2m} P(i,i) + \sum_{i=m+1}^{p} \frac{mn}{8\tilde{\tau}^2} P(i,i) \\
&\leq\; \frac{1}{2}\beta + \frac{mn}{8\tilde{\tau}},
\end{aligned}$$

since $P(i,i) \leq \beta$ for all $i$ and $\mathrm{Tr}(\mathbf{P}) \leq \tilde{\tau}$. We also have

$$\begin{aligned}
\mathbf{Y} \bullet \mathsf{sym}(\mathbf{W}) &= \mathrm{Tr}(\mathbf{D}\mathbf{U}\mathbf{U}^\top \mathbf{D}\mathsf{sym}(\mathbf{W})) \\
&= \mathrm{Tr}(\mathbf{U}\mathbf{U}^\top \mathbf{D}\mathsf{sym}(\mathbf{W})\mathbf{D}) \\
&= \frac{\sqrt{n}}{4\tilde{\tau}}\mathrm{Tr}(\mathbf{U}\mathbf{U}^\top \mathsf{sym}(\mathbf{W})) && (\because \mathbf{D}\mathsf{sym}(\mathbf{W})\mathbf{D} = \mathsf{sym}(\tfrac{\sqrt{n}}{4\tilde{\tau}}\mathbf{W})) \\
&= \frac{\sqrt{n}}{4\tilde{\tau}}\mathrm{Tr}(\mathbf{U}\mathbf{U}^\top [\mathbf{U}(\sqrt{n}\mathbf{I}_m)\mathbf{U}^\top + \mathbf{V}(-\sqrt{n}\mathbf{I}_m)\mathbf{V}^\top]) \\
&= \frac{mn}{4\tilde{\tau}},
\end{aligned}$$

since $\mathbf{U}^\top\mathbf{V} = \mathbf{0}$. Putting the above two inequalities together, we have

$$\frac{mn}{4\tilde{\tau}} \leq \frac{1}{2}\beta + \frac{mn}{8\tilde{\tau}},$$

which implies that

$$\beta\tilde{\tau} \geq \frac{1}{4}mn = \frac{1}{4}\tau\sqrt{n}$$

as required. □

# E Relation between $(\beta,\tau)$-decomposition, max-norm and trace-norm

In this section, we consider $m \times n$ non-symmetric matrix $\mathbf{W}$. The max-norm of $\mathbf{W}$ is defined to be (see Lee et al. [2010]) the value of the following SDP:

$$\begin{aligned}
&\min t \\
&\begin{bmatrix} \mathbf{Y}_1 & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{Y}_2 \end{bmatrix} \succeq \mathbf{0} \\
\forall i \in [m], j \in [n]: \; &Y_1(i,i),\; Y_2(j,j) \leq t. \qquad (8)
\end{aligned}$$

The least possible $\beta$ in any $(\beta,\tau)$-decomposition for $\mathbf{W}$ is given by the following SDP:

$$\begin{aligned}
&\min \beta \\
&\begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{0} \end{bmatrix} = \mathbf{P} - \mathbf{N} \\
&\mathbf{P},\; \mathbf{N} \succeq \mathbf{0} \\
\forall i \in [m+n]: \; &P(i,i),\; N(i,i) \leq \beta. \qquad (9)
\end{aligned}$$

**Theorem 21.** *The least possible $\beta$ in any $(\beta,\tau)$-decomposition exactly equals half the max-norm of $\mathbf{W}$.*

*Proof.* Let $t^*$ and $\beta^*$ be the optima of SDPs (8) and (9) respectively. Let $\mathbf{Y}_1,\; \mathbf{Y}_2$ be the optimal solution to SDP (8), so that for all $i \in [m],\; j \in [n]$ we have $Y_1(i,i),\; Y_2(j,j) \leq t^*$. Consider the matrices

$$\mathbf{P} = \frac{1}{2}\begin{bmatrix} \mathbf{Y}_1 & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{Y}_2 \end{bmatrix} \text{ and } \mathbf{N} = \frac{1}{2}\begin{bmatrix} \mathbf{Y}_1 & -\mathbf{W} \\ -\mathbf{W}^\top & \mathbf{Y}_2 \end{bmatrix}.$$

Using the feasibility of $\mathbf{Y}_1$, $\mathbf{Y}_2$ and Lemma 22, we get that $\mathbf{P}, \mathbf{N} \succeq \mathbf{0}$. Thus this is a feasible solution to SDP (9). Hence, we conclude that $t^* \geq 2\beta^*$.

Now let $\mathbf{P}$, $\mathbf{N}$ be the optimal solution to SDP (9), so that for all $i \in [m+n]$ we have $P(i,i),\ N(i,i) \leq \beta^*$. Consider the blocks of $\mathbf{P}$ and $\mathbf{N}$ formed by the first $m$ indices and the last $n$ indices:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^A & \mathbf{P}^B \\ \mathbf{P}^C & \mathbf{P}^D \end{bmatrix} \text{ and } \mathbf{N} = \begin{bmatrix} \mathbf{N}^A & \mathbf{N}^B \\ \mathbf{N}^C & \mathbf{N}^D \end{bmatrix}.$$

Since $\mathbf{N} \succeq \mathbf{0}$, by Lemma 22 the following matrix is positive semidefinite as well:

$$\mathbf{N}' \ := \ \begin{bmatrix} \mathbf{N}^A & -\mathbf{N}^B \\ -\mathbf{N}^C & \mathbf{N}^D \end{bmatrix} \ \succeq \ \mathbf{0}.$$

So $\mathbf{P} + \mathbf{N}' \succeq \mathbf{0}$, i.e.

$$\mathbf{P} + \mathbf{N}' \ = \ \begin{bmatrix} \mathbf{P}^A + \mathbf{N}^A & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{P}^D + \mathbf{N}^D \end{bmatrix} \ \succeq \ \mathbf{0}.$$

Thus, $\mathbf{Y}_1 = \mathbf{P}^A + \mathbf{N}^A$ and $\mathbf{Y}_2 = \mathbf{P}^D + \mathbf{N}^D$ is a feasible solution to SDP (8). Now for all $i \in [m]$ we have $Y_1(i,i) \leq P^A(i,i) + N^A(i,i) \leq 2\beta^*$, and similarly for all $j \in [n]$ we have $Y_2(j,j) \leq 2\beta^*$. Thus, we conclude that $t^* \leq 2\beta^*$. □

**Lemma 22.** *Let $\mathbf{P}$ be a positive semidefinite matrix of order $m+n$ and let*

$$\mathbf{P} \ = \ \begin{bmatrix} \mathbf{P}^A & \mathbf{P}^B \\ \mathbf{P}^C & \mathbf{P}^D \end{bmatrix}.$$

*be the block decomposition of $\mathbf{P}$ formed by the first $m$ indices and the last $n$ indices. Then the following matrix is positive semidefinite:*

$$\mathbf{P'} \ := \ \begin{bmatrix} \mathbf{P}^A & -\mathbf{P}^B \\ -\mathbf{P}^C & \mathbf{P}^D \end{bmatrix}.$$

*Proof.* Since $\mathbf{P} \succeq \mathbf{0}$, there are vectors $\mathbf{v}_i$, for all $i, j \in [m+n]$ such that $P(i,j) = \mathbf{v}_i \cdot \mathbf{v}_j$. Then consider the vectors

$$\mathbf{w}_i \ := \ \begin{cases} \mathbf{v}_i & \text{if } i \in [m] \\ -\mathbf{v}_i & \text{otherwise.} \end{cases}$$

It is easy to check that for all $i, j \in [m+n]$ we have $P'(i,j) = \mathbf{w}_i \cdot \mathbf{w}_j$. Thus, we conclude that $\mathbf{P'} \succeq \mathbf{0}$. □

Finally, we show the connection between the trace-norm and the least possible $\tau$ in any $(\beta, \tau)$-decomposition:

**Theorem 23.** *The least possible $\tau$ in any $(\beta, \tau)$-decomposition exactly equals twice the trace-norm of $\mathbf{W}$.*

*Proof.* Let $\tau^*$ be the least possible value of $\tau$ in any $(\beta, \tau)$-decomposition, and let $\mathbf{P}, \mathbf{N}$ be positive semidefinite matrices such that $\mathsf{sym}(\mathbf{W}) = \mathbf{P} - \mathbf{N}$ and $\mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) = \tau^*$. Then by triangle inequality, we have

$$\|\mathsf{sym}(\mathbf{W})\|_\star \ \leq \ \|\mathbf{P}\|_\star + \|\mathbf{N}\|_\star.$$

Since $\|\mathsf{sym}(\mathbf{W})\|_\star = 2\|\mathbf{W}\|_\star$, $\|\mathbf{P}\|_\star = \mathrm{Tr}(\mathbf{P})$, and $\|\mathbf{N}\|_\star = \mathrm{Tr}(\mathbf{N})$, we conclude that $\tau^* \geq 2\|\mathbf{W}\|_\star$. Now, let

$$\mathsf{sym}(\mathbf{W}) = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$$

be the eigenvalue decomposition of $\mathsf{sym}(\mathbf{W})$. Now consider the positive semidefinite matrices

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

Clearly $\mathsf{sym}(\mathbf{W}) = \mathbf{P} - \mathbf{N}$, and

$$\mathrm{Tr}(\mathbf{P}) + \mathrm{Tr}(\mathbf{N}) \ = \ \sum_i |\lambda_i| \ = \ \|\mathsf{sym}(\mathbf{W})\|_\star \ = \ 2\|\mathbf{W}\|_\star.$$

Hence, $\tau^* \leq 2\|\mathbf{W}\|_\star$, completing the proof. □