Follow the Compressed Leader: Faster Online Learning of Eigenvectors and Faster MMWU

Zeyuan Allen-Zhu [*1] **Yuanzhi Li** [*2]

Abstract

The online problem of computing the top eigenvector is fundamental to machine learning. The famous matrix-multiplicative-weight-update (MMWU) framework solves this online problem and gives optimal regret. However, since MMWU runs very slow due to the computation of matrix exponentials, researchers proposed the follow-the-perturbed-leader (FTPL) framework which is faster, but a factor $\sqrt{d}$ worse than the optimal regret for dimension-d matrices. We propose a *follow-the-compressed-leader* framework which, not only matches the optimal regret of MMWU (up to polylog factors), but runs no slower than FTPL. Our main idea is to "compress" the MMWU strategy to dimension 3 in the adversarial setting, or dimension 1 in the stochastic setting. This resolves an open question regarding how to obtain both (nearly) optimal and efficient algorithms for the online eigenvector problem.

1 Introduction

Finding leading eigenvectors of symmetric matrices is one of the most primitive problems in machine learning. In this paper, we study the *online* variant of this problem, which is a learning game between a player and an adversary (Nie et al., 2013; Kotłowski & Warmuth, 2015; Dwork et al., 2014; Garber et al., 2015; Abernethy et al., 2015).

Online Eigenvector Problem. The player plays T unit-norm vectors $w_1, \dots, w_T \in \mathbb{R}^d$ in a row; after playing w_k, the adversary picks a feedback matrix $\mathbf{A}_k \in \mathbb{R}^{d\times d}$ that is symmetric and satisfies $0 \preceq \mathbf{A}_k \preceq \mathbf{I}$.[1] Both these assumptions are for the sake of simplicity and can be relaxed.[2] The player then receives a gain

$$w_k^\top \mathbf{A}_k w_k = \mathbf{A}_k \bullet w_k w_k^\top \in [0,1] \ .$$

The regret minimization problem asks us the player to design a strategy to minimize *regret*, that is, the difference between the total gain obtained by the player and that by the *a posteriori* best fixed strategy $u \in \mathbb{R}^d$:

$$\boxed{\begin{aligned}\text{minimize}\quad & \max_{u\in\mathbb{R}^d} \textstyle\sum_{k=1}^T \mathbf{A}_k \bullet (uu^\top - w_k w_k^\top) \\ & = \lambda_{\max}(\mathbf{A}_1 + \cdots + \mathbf{A}_T) - \sum_{k=1}^T w_k^\top \mathbf{A}_k w_k \ .\end{aligned}}$$

The name comes from the fact that the player chooses only vectors in a row, but wants to compete against the leading eigenvector in hindsight. To make this problem meaningful, the feedback matrix $\mathbf{A}_k$, is *not* allowed to depend on w_k but can depend on $w_1, \dots, w_{k-1}$.

1.1 Known Results

The most famous solution to the online eigenvector problem is the *matrix multiplicative-weight-update (MMWU)* method, which has also been used towards efficient algorithms for SDP, balanced separators, Ramanujan sparsifiers, and even in the proof of QIP $=$ PSPACE.

MMWU. At iteration k, define $\mathbf{W}_k = \frac{\exp(\eta \mathbf{\Sigma}_{k-1})}{\mathbf{Tr}\exp(\eta\mathbf{\Sigma}_{k-1})}$ where $\mathbf{\Sigma}_{k-1} := \mathbf{A}_1 + \cdots + \mathbf{A}_{k-1}$ and $\eta > 0$ is the learning rate. Then, compute its eigendecomposition

$$\mathbf{W}_k = \frac{\exp(\eta \mathbf{\Sigma}_{k-1})}{\mathbf{Tr}\exp(\eta\mathbf{\Sigma}_{k-1})} = \textstyle\sum_{j=1}^d p_j \cdot y_j y_j^\top$$

where vectors y_j are normalized eigenvectors. Now, the MMWU strategy instructs the player to choose $w_k = y_j$

[*]Equal contribution . Future version of this paper shall be found at `https://arxiv.org/abs/1701.01722`. [1]Microsoft Research [2]Princeton University. Correspondence to: Zeyuan Allen-Zhu <zeyuan@csail.mit.edu>, Yuanzhi Li <yuanzhil@cs.princeton.edu>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1]We denote by $\mathbf{A} \succeq \mathbf{B}$ spectral dominance that is equivalent to saying that $\mathbf{A} - \mathbf{B}$ is positive semidefinite (PSD).

[2]Firstly, all the results cited and stated in this paper, after scaling, generalize to the scenario when the eigenvalues of $\mathbf{A}_k$ are in the range $[l, r]$ for arbitrary $l, r \in \mathbb{R}$. For notational simplicity, we have assumed $l = 0$ and $r = 1$ in this paper. Secondly, if $\mathbf{A}_k$ is not symmetric or even rectangular, classical reductions can turn such a problem into an equivalent online game with only symmetric matrices (see Sec 2.1 of (Garber et al., 2015)).

each with probability p_j. The best choice $\eta = \sqrt{\log d}/\sqrt{T}$ yields a total expected regret $O(\sqrt{T \log d})$ (Orecchia, 2011), and this is optimal up to constant (Arora et al., 2012). It requires some additional, but standard, effort to turn this into a high-confidence result.

Unfortunately, the per-iteration running time of MMWU is at least $O(d^\omega)$ due to eigendecomposition, where d^ω is the complexity for multiplying two $d \times d$ matrices.[3]

MMWU-JL. Some researchers also use the Johnson-Lindenstrauss (JL) compression to reduce the dimension of $\mathbf{W}_k$ from MMWU to make it more efficiently computable (Peng & Tangwongsan, 2012; Allen-Zhu et al., 2016; 2015; Lee & Sun, 2015). Specifically, they compute a sketch matrix $\mathbf{Y} = \mathbf{W}_k^{1/2}\mathbf{Q}$ using a random $\mathbf{Q} \in \mathbb{R}^{d\times m}$, and then use $\mathbf{Y}\mathbf{Y}^\top$ to approximate $\mathbf{W}_k$. If the dimension m is $\widetilde{O}(1/\sigma^2)$, this compression incurs an average regret loss of σ. We call this method MMWU-JL for short.[4]

Unfortunately, to maintain a total regret $\widetilde{O}(\sqrt{T})$, one must let $\sigma \approx T^{-1/2}$. Therefore, JL compresses the matrix exponential to dimension $\widetilde{O}(T)$, and is only useful when $T \le d$.

FTPL. Researchers also study the *follow-the-perturbed-leader (FTPL)* strategy (Kotłowski & Warmuth, 2015; Dwork et al., 2014; Garber et al., 2015; Abernethy et al., 2015). Most notably, Garber, Hazan and Ma (Garber et al., 2015) proposed to compute an (approximate) leading eigenvector of the matrix $\mathbf{\Sigma}_{k-1} + rr^\top$ at iteration k, where r is a random vector whose norm is around $\sqrt{dT}$.

Unfortunately, the total regret of FTPL is $\widetilde{O}(\sqrt{dT})$, which is a factor $\sqrt{d}$ worse than the optimum regret, and interesting only when $T \ge d$. This factor $\sqrt{d}$ loss can indeed be realized in practice, see Figure 1.

1.2 Our Main Results

We propose a *follow-the-compressed-leader (FTCL)* strategy that, at a high level, compresses the MMWU strategy to dimension $m = 3$ as opposed to dimension $m = \widetilde{\Theta}(T)$ in MMWU-JL. Our FTCL strategy has significant advantages over previous results because:

- FTCL has regret $\widetilde{O}(\sqrt{T})$ which is optimal up to poly-log factors (as opposed to $\sqrt{d}$ in FTPL).
- Each iteration of FTCL is dominated by solving a logarithmic number of linear systems.

Since solving linear systems is generally no slower than computing eigenvectors or matrix exponentials, the per-iteration complexity of FTCL is no slower than FTPL, and much faster than MMWU and MMWU-JL. We shall make this comparison more explicit in Section 3.

1.3 Our Side Result: Stochastic Online Eigenvector

We also study the *special case* of the online eigenvector problem where the adversary is *stochastic*, meaning that $\mathbf{A}_1, \dots, \mathbf{A}_T$ are chosen i.i.d. from a common distribution whose expectation equals some matrix $\mathbf{B}$, independent of the player's actions. For this problem,

- Garber *et al.* (Garber et al., 2015) showed a block power method gives a total regret $O(\sqrt{T \log(dT)})$, and runs in $O(\mathsf{nnz}(\mathbf{\Sigma}_T))$ time per iteration. (We denote $\mathsf{nnz}(\mathbf{M})$ the time to multiply $\mathbf{M}$ to a vector.)
- Shamir (Shamir, 2016) showed Oja's algorithm[5] has a total regret $O(\sqrt{dT}\log(T))$, which is a factor $\sqrt{d}$ worse than optimum.

In this paper, we show that Oja's algorithm in fact only has a total regret $O(\sqrt{T}\log d)$ for this stochastic setting, which is optimal up to a $\sqrt{\log d}$ factor. Most importantly, the k-th iteration of Oja's runs in only $O(\mathsf{nnz}(\mathbf{A}_k))$ time.

Example. Since in low-rank or sparse cases it usually satisfies $\mathsf{nnz}(\mathbf{\Sigma}_T) = d^2$ and $\mathsf{nnz}(\mathbf{A}_k) = O(d)$, our result can be faster than block power method by a factor $O(d)$.

Our proof relies on a compression view of Oja's algorithm which compresses MMWU to dimension $m = 1$. Our proof is one-paged, indicating that FTCL might be a better framework of designing online algorithms for matrices.

1.4 Our Results in a More Refined Language

Denoting by $\lambda := \frac{1}{T}\lambda_{\max}(\mathbf{A}_1 + \cdots \mathbf{A}_T)$, we have $\lambda \le 1$ according to the normalization $\mathbf{A}_k \preceq \mathbf{I}$. In general, the smaller λ is, the better a learning algorithm should behave. In the previous subsections, we have followed the tradition and discussed our results and prior works assuming the *worst* possibility of λ. This has indeed simplified notations.

If λ is much smaller than 1, our complexity bounds can be improved to quantities that depend on λ. We call this the λ-*refined language*. At a high level, for our FTCL, in both the adversarial and stochastic settings, the total regret improves from $\widetilde{O}(\sqrt{T})$ to $\widetilde{O}(\sqrt{\lambda T})$.

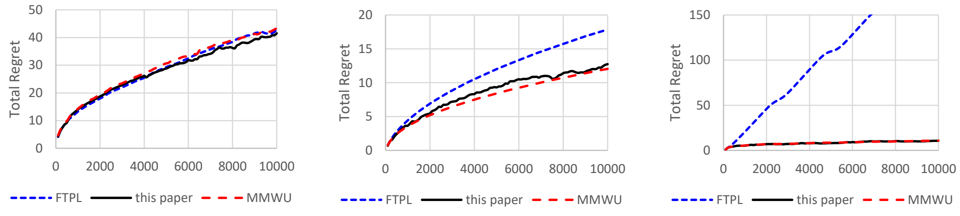
We have an information-theoretic lower bound of $\Omega(\sqrt{\lambda T})$ for the total regret in this λ-refined language, see full version. This lower bound even holds for the stochastic problem, even when the matrices $\mathbf{A}_k$ are of rank 1.

As for prior work, it has been recorded that (cf. Theorem 3.1 of (Allen-Zhu et al., 2015)) the MMWU and MMWU-JL methods have total regret $O(\sqrt{\lambda T \log d})$. The block

[3]In fact, it is known that eigendecomposition has complexity $O(d^\omega)$ when all the eigenvalues are distinct, and could possibly go up to $O(d^3)$ when some eigenvalues are equal (Pan & Chen, 1999).

[4]Through the paper, we use the $\widetilde{O}$ notation to hide polylogarithmic factors in T, d and $1/\varepsilon$ if applicable.

[5]Here is a simple description of Oja's algorithm: beginning with a random Gaussian vector $u \in \mathbb{R}^d$, at each iteration k, choose w_k to be $(\mathbf{I}+\eta\mathbf{A}_{k-1})\cdots(\mathbf{I}+\eta\mathbf{A}_1)u$ after normalization.



(a) `random` (b) `diagonal` (c) `diagonal+rotation`

Figure 1: We generate synthetic data to verify that the total regret of FTPL can indeed be poorer than MMWU or our FTCL. We explain how matrices $\mathbf{A}_k$ are chosen in Appendix A. We have $d = 100$ and the x-axis represents the number of iterations.

Paper	**Total Regret**	**Time Per Iteration**	**Minimum Total Time for ε Average Regret**[a]
MMWU	$\widetilde{O}(\sqrt{T})$	at least $O(d^\omega)$	$\widetilde{O}\big(\frac{d^\omega}{\varepsilon^2}\big)$
MMWU-JL($T \leq d$ only)	$\widetilde{O}(\sqrt{T})$	$\mathsf{M}^{\mathsf{exp}} \times \widetilde{O}(T)$	$\widetilde{O}\big(\frac{1}{\varepsilon^{4.5}}\mathsf{nnz}(\mathbf{\Sigma})\big)$
FTPL ($T \geq d$ only)	$\widetilde{O}(\sqrt{dT})$	$\mathsf{M}^{\mathsf{ev}} \times 1$	$\widetilde{O}\big(\frac{d^{1.5}}{\varepsilon^{3.5}}\mathsf{nnz}(\mathbf{\Sigma})\big)$
this paper	$\widetilde{O}(\sqrt{T})$ Theorem 1&2	$\mathsf{M}^{\mathsf{lin}} \times \widetilde{O}(1)$ Theorem 3	$\widetilde{O}\big(\frac{1}{\varepsilon^{2.5}}\mathsf{nnz}(\mathbf{\Sigma})\big)$ and $\widetilde{O}\big(\frac{1}{\varepsilon^{2.5}}\mathsf{nnz}(\mathbf{\Sigma})^{\frac{3}{4}}\mathsf{nnz}(\mathbf{A})^{\frac{1}{4}} + \frac{1}{\varepsilon^2}\mathsf{nnz}(\mathbf{\Sigma})\big)$
$\downarrow$ stochastic online eigenvector only $\downarrow$			
block power method	$\widetilde{O}(\sqrt{T})$	$O\big(\mathsf{nnz}(\mathbf{\Sigma})\big)$	$\widetilde{O}\big(\frac{1}{\varepsilon^2}\mathsf{nnz}(\mathbf{\Sigma})\big)$
this paper	$\widetilde{O}(\sqrt{T})$ Theorem 4	$O\big(\mathsf{nnz}(\mathbf{A})\big)$ Theorem 4	$\widetilde{O}\big(\frac{1}{\varepsilon^2}\mathsf{nnz}(\mathbf{A})\big)$

Table 1: Comparison of known methods for the online eigenvector problem. We denote by $\mathsf{nnz}(\mathbf{M})$ the time needed to multiply $\mathbf{M}$ to a vector, by $\mathbf{\Sigma} = \mathbf{A}_1 + \cdots + \mathbf{A}_T$, and by $\mathsf{nnz}(\mathbf{A}) = \max_{k\in[T]}\big\{\mathsf{nnz}(\mathbf{A}_k)\big\} \leq \mathsf{nnz}(\mathbf{\Sigma})$.

- $\mathsf{M}^{\mathsf{exp}}$ is the time to compute $e^{-\mathbf{M}}$ multiplied with a vector, where $\mathbf{M} \in \mathbb{R}^{d\times d}$ satisfies $0 \preceq \mathbf{M} \preceq \widetilde{O}(T^{1/2}) \cdot \mathbf{I}$.
- M^{ev} is the time to compute the leading eigenvector of matrix $\mathbf{M}$ to multiplicative accuracy $O(T^{-3/2}d^{1/2}) \in (0,1)$.
- $\mathsf{M}^{\mathsf{lin}}$ is the time to solve a linear system for matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$, where $\mathbf{M}$ is PSD and of condition number $\leq \widetilde{O}(T^{1/2})$.
- If using iterative methods, the worst-case values M^{ev}, $\mathsf{M}^{\mathsf{exp}}$, $\mathsf{M}^{\mathsf{lin}}$ are

$$\mathsf{M}^{\mathsf{ev}} = \widetilde{O}\big(\min\{T^{\frac{3}{4}}d^{-\frac{1}{4}}\mathsf{nnz}(\mathbf{\Sigma}), d^\omega\}\big) \geq \mathsf{M}^{\mathsf{exp}} = \widetilde{O}\big(\min\{T^{\frac{1}{4}}\mathsf{nnz}(\mathbf{\Sigma}), d^\omega\}\big) \geq \mathsf{M}^{\mathsf{lin}} = \widetilde{O}\big(\min\{\min\{d, T^{\frac{1}{4}}\}\mathsf{nnz}(\mathbf{\Sigma}), d^\omega\}\big) ,$$

where d^ω is the time needed to multiply two $d \times d$ matrices. If using stochastic iterative methods, $\mathsf{M}^{\mathsf{lin}}$ is at most $\widetilde{O}\big(T^{\frac{1}{4}}\mathsf{nnz}(\mathbf{\Sigma})^{\frac{3}{4}}\mathsf{nnz}(\mathbf{A})^{\frac{1}{4}} + \mathsf{nnz}(\mathbf{\Sigma})\big)$.

[a]The total time complexity of the first T_ε rounds where T_ε is the earliest round to achieve an ε average regret.

power method (for the stochastic setting) has total regret $\widetilde{O}(\sqrt{\lambda T})$, by modifying the proof in (Garber et al., 2015). To the best of our knowledge, FTPL has not been analyzed in the λ-refined language. We compare with prior work in Table 2 in the appendix for this λ-refined language.

1.5 Other Related Works

The multiplicative weight update (MWU) method is a simple but extremely powerful algorithmic tool that has been repeatedly discovered in theory of computation, machine learning, optimization, and game theory (see for instance the survey (Arora et al., 2012) and the book (Cesa-Bianchi & Lugosi, 2006)). Its natural matrix extension, matrix-multiplicative-weight-update (MMWU) (Orecchia, 2011), has been used towards efficient algorithms for solving semidefinite programs (Arora & Kale, 2007; Allen-Zhu et al., 2016; Peng & Tangwongsan, 2012), balanced separators (Orecchia et al., 2012), Ramanujan sparsifiers (Allen-Zhu et al., 2015; Lee & Sun, 2015), and even in the proof of QIP = PSPACE (Jain et al., 2011). Some authors also refer to MMWU as the *follow-the-regularized-leader* strategy or FTRL for short, because MMWU can be analyzed from a mirror-descent view with the matrix entropy function as its regularizer.

For the online eigenvector problem, if the feedback matrices $\mathbf{A}_k$ are only of rank-1, the $\widetilde{O}(\sqrt{dT})$ total regret of FTPL can be improved to $\widetilde{O}(d^{1/4}T^{1/2})$. This is first shown by Dwork *et al.* (Dwork et al., 2014) and independently by Kotłowski and Warmuth (Kotłowski & Warmuth, 2015). Abernethy *et al.* showed FTPL strategies can be analyzed using a FTRL framework (Abernethy et al., 2014).

Researchers also put efforts to understand high-rank variants of the online eigenvector problem. Nie *et al.* studied the high-rank variant using MMWU (Nie et al., 2013), but their per-iteration complexity is also high due to eigendecomposition. Some authors study a very different online model for computing the top k eigenvectors (Boutsidis et al., 2015; Karnin & Liberty, 2015): they wish to output $O(k \cdot \mathsf{poly}(1/\varepsilon))$ vectors instead of k but with a good PCA reconstruction error.

The *stochastic* online eigenvector problem is related but different from streaming PCA (Hardt & Price, 2014; Allen-Zhu & Li, 2016b). In streaming PCA, we are given i.i.d. random matrices with an expectation $\mathbf{B}$ and asked to find a unit vector w with large $w^\top \mathbf{B} w$ in the end, without worrying about the per-iteration gain. The cited papers use different techniques from ours and do not imply our result on stochastic online eigenvector.

For the most efficient offline eigenvectors algorithms, we refer interested readers to our paper (Allen-Zhu & Li, 2016a) (for PCA / SVD) and (Allen-Zhu & Li, 2017) (for CCA and generalized eigendecomposition).

1.6 Roadmap

We introduce notations in Section 2, and compare the per-iteration complexity of FTCL to prior work in Section 3. We discuss high-level intuitions and techniques in Section 4. We introduce a new trace inequality in Section 5, and state our main FTCL result for an oblivious adversary in Section 6. We extend it to the adversarial setting and discuss how to implement FTCL fast in the full version of this paper. Finally, in Section 8 we provide our FTCL result for a stochastic adversary.

Our results are stated directly in the λ-refined language.

2 Notations and Preliminaries

Define $\mathbf{\Sigma}_k := \sum_{i=1}^k \mathbf{A}_i$ for every $k = 0, 1, \dots, T$. Since each $\mathbf{A}_k$ is positive semi-definite (PSD), we can find $\mathbf{P}_k \in \mathbb{R}^{d\times d}$ such that $\mathbf{A}_k = \mathbf{P}_k \mathbf{P}_k^\top$; we only use $\mathbf{P}_k$ for analysis purpose only. Given two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times d}$, we write $\mathbf{A} \bullet \mathbf{B} := \mathbf{Tr}(\mathbf{A}^\top \mathbf{B})$. We write $\mathbf{A} \succeq \mathbf{B}$ if $\mathbf{A}, \mathbf{B}$ are symmetric matrices and $\mathbf{A} - \mathbf{B}$ is PSD. We write $[\mathbf{A}]_{i,j}$ the (i,j)-th entry of $\mathbf{A}$. We use $\|\mathbf{M}\|_2$ to denote the spectral norm of a matrix $\mathbf{M}$. We use $\mathsf{nnz}(\mathbf{M})$ to denote time needed to multiply matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$ with an arbitrary vector in $\mathbb{R}^d$. In particular, $\mathsf{nnz}(\mathbf{M})$ is at most d plus the number of non-zero elements in $\mathbf{M}$. We denote $\mathsf{nnz}(\mathbf{A}) := \max_{k\in[T]} \{\mathsf{nnz}(\mathbf{A}_k)\}$.

Suppose $x_1, \cdots, x_t \in \mathbb{R}$ are drawn i.i.d. from the standard Gaussian $\mathcal{N}(0,1)$, then $\chi = \sum_{i=1}^t x_i^2$ has a chi-squared distribution of t-degree freedom. χ^{-1} is called inverse-chi-squared distribution of t-degree freedom. It is known that $\mathbb{E}[\chi^{-1}] = \frac{1}{t-2}$ for $t \geq 3$.

3 Detailed Comparison to Prior Work

We compare the per-iteration complexity of our results more closely to prior work.

In the stochastic setting, Oja's method runs in time $\mathsf{nnz}(\mathbf{A}_k)$ for iteration k, and therefore is *clearly faster* than the block power method which runs in time $\mathsf{nnz}(\mathbf{\Sigma}_k)$.

In the adversarial setting, it is *clear* that the per-iteration complexities of FTPL and FTCL are no greater than MMWU, because computing the leading eigenvector and the matrix inversion are both faster than computing the full eigendecomposition. In the rest of this section, we compare MMWU-JL, FTPL and FTCL more closely. They respectively have per-iteration complexities

$$\widetilde{O}(T) \times \mathsf{M}^{\mathsf{exp}}, \quad 1 \times \mathsf{M}^{\mathsf{ev}}, \quad \text{and} \quad \widetilde{O}(1) \times \mathsf{M}^{\mathsf{lin}}$$

where

- In MMWU-JL, we denote by $\mathsf{M}^{\mathsf{exp}}$ the time needed for computing $\exp(\eta \mathbf{\Sigma}_{k-1}/2)$ multiplied to a vector. Recall that $\eta = \widetilde{\Theta}(T^{-1/2})$.
- In FTPL, following the tradition, we denote by M^{ev} the time needed for computing the top eigenvector of $\mathbf{\Sigma}_{k-1} + rr^\top$, where the norm of r is $O(\sqrt{dT})$.
- In FTCL, we denote by $\mathsf{M}^{\mathsf{lin}}$ the time needed for solving a linear system with matrix $\mathbf{M} = c\mathbf{I} - \eta\mathbf{\Sigma}_{k-1}$, where $\mathbf{M} \succeq \frac{1}{e}\mathbf{I}$ and $\eta = \widetilde{\Theta}(T^{-1/2})$.

For exact computations, one may generally derive that $\mathsf{M}^{\mathsf{exp}} \geq \mathsf{M}^{\mathsf{ev}} \geq \mathsf{M}^{\mathsf{lin}}$. However, for large-scale applications, one usually applies iterative methods for the three tasks. Iterative methods utilize matrix sparsity, and have running times that depend on matrix properties.

Worst-case Complexity. We compute that:

- $\mathsf{M}^{\mathsf{exp}}$ in the worst case is $\widetilde{O}(\min\{T^{1/4}\mathsf{nnz}(\mathbf{\Sigma}_T), d^\omega\})$.

  The first is because if using Chebyshev approximation, one can compute $\exp(\eta\mathbf{\Sigma}_{k-1}/2)$ applied to a vector in time at most $\widetilde{O}\big(\|\eta\mathbf{\Sigma}_{k-1}\|_2^{1/2} \cdot \mathsf{nnz}(\mathbf{\Sigma}_{k-1})\big)$. The second is because one can compute the singular value decomposition of $\mathbf{\Sigma}_{k-1}$ in time $\widetilde{O}(d^\omega)$ and then compute the matrix $\exp(\eta\mathbf{\Sigma}_{k-1}/2)$ directly.
- M^{ev} in the worst case is $\widetilde{O}(\min\{T^{3/4}d^{-1/4}\mathsf{nnz}(\mathbf{\Sigma}_T), d^\omega\})$.

  The first is so because, as proved in (Garber et al., 2015), it suffices to compute the top eigenvector of $\mathbf{\Sigma}_{k-1} + rr^\top$ up to a multiplicative error $O(T^{-\frac{3}{2}}d^{\frac{1}{2}})$.[6] If one applies Lanczos method, this is in time $\widetilde{O}\big(T^{\frac{3}{4}}d^{-\frac{1}{4}}\mathsf{nnz}(\mathbf{\Sigma}_T)\big)$. (Recall that it only works when $T \geq d$). The second is because the leading eigenvector of a $d\times d$ matrix can be computed directly in time $O(d^\omega)$.

[6] A multiplicative error δ means to find x such that $x^\top(\mathbf{\Sigma}_{k-1} + rr^\top)x \geq (1-\delta)\lambda_{\max}(\mathbf{\Sigma}_{k-1} + rr^\top)$.

- $\mathsf{M}^{\mathsf{lin}}$ in the worst case is $\widetilde{O}\big(\min\{\min\{T^{\frac{1}{4}}, d\}\mathsf{nnz}(\mathbf{\Sigma}_T), d^{\omega}\}\big)$.

  The first is because our matrix $\mathbf{M}$ has a condition number at most $O(\eta T) = \widetilde{O}(T^{1/2})$. If using conjugate gradient (Shewchuk, 1994), one can solve a linear system for $\mathbf{M}$ in time at most $\widetilde{O}\big(\min\{T^{\frac{1}{4}}, d\}\mathsf{nnz}(\mathbf{\Sigma}_T)\big)$. The second is because the inverse of a $d \times d$ matrix can be computed directly in time $O(d^{\omega})$ (Bunch & Hopcroft, 1974).

- $\mathsf{M}^{\mathsf{lin}}$ can be improved to $\widetilde{O}\big(\min\big\{T^{\frac{1}{4}}\mathsf{nnz}(\mathbf{\Sigma}_T)^{\frac{3}{4}}\mathsf{nnz}(\mathbf{A})^{\frac{1}{4}} + \mathsf{nnz}(\mathbf{\Sigma}_T), d^{\omega}\big\}\big)$ if using stochastic iterative methods.

In sum, if using iterative methods, the worst case values of $\mathsf{M}^{\mathsf{lin}}$, M^{ev}, $\mathsf{M}^{\mathsf{exp}}$ are on the same magnitude. Since the per-iteration cost of FTCL is only $\widetilde{O}(\mathsf{M}^{\mathsf{lin}})$, this is no slower than $O(\mathsf{M}^{\mathsf{ev}})$ of FTPL, and much faster than $O(T \times \mathsf{M}^{\mathsf{exp}})$ of MMWU-JL.

Practical Complexity. There are many algorithms to compute leading eigenvectors, including Lanczos method, shift-and-invert, and the (slower) power method. The performance may depend on other properties of the matrix, including "how well-clustered the eigenvalues are."

There are also numerous ways to compute matrix inversions, including conjugate gradient, accelerated coordinate descent, Chebyshev method, accelerated SVRG, and many others. Some of them also run faster when the eigenvalues form clusters (Shewchuk, 1994).

In particular, for a random Gaussian matrix $\mathbf{\Sigma}_{k-1}$ (with dimension $100 \sim 5000$), using the default scientific package SciPy of Python, M^{ev} is roughly 3 times of $\mathsf{M}^{\mathsf{lin}}$.

Total Worst-Case Complexity. Since FTPL requires d times *more iterations* in order to achieve the same average regret as FTCL or MMWU, in the last column of Table 1, we also summarize the minimum *total* time complexity needed to achieve an ε average regret.

Examples. If $\mathsf{nnz}(\mathbf{\Sigma}_T) = d^2$ and $\mathsf{nnz}(\mathbf{A}) = O(d)$, the total complexity needed to achieve an ε average regret:

$\widetilde{O}(d^2\varepsilon^{-2} + d^{1.75}\varepsilon^{-2.5})$ (us) $\quad \widetilde{O}(d^{3.5}\varepsilon^{-3.5})$ (FTPL)

$\widetilde{O}(d^2\varepsilon^{-4.5})$ (MMWU-JL) $\quad \widetilde{O}(d^3\varepsilon^{-2})$ (MMWU)

In the λ-refined setting, one can revise the complexity bounds accordingly. We ignore the details in this short version and present them in Table 2 in the appendix.

4 High-Level Discussion of Our Techniques

Revisit MMWU. We first revisit the high-level idea behind the proof of MMWU. Recall $\mathbf{W}_k = \exp(c_k\mathbf{I} + \eta\mathbf{\Sigma}_{k-1})$ where c_k is the unique constant such that $\mathbf{Tr}\mathbf{W}_k = 1$. The main proof step (see for instance Theorem 3.1 of (Allen-Zhu et al., 2015)) is to use the equality $\mathbf{Tr}\mathbf{W}_k = \mathbf{Tr}\mathbf{W}_{k+1} = 1$ to derive a relationship between $c_k - c_{k+1}$ and the gain value $\mathbf{W}_k \bullet \mathbf{A}_k$ at this iteration.

More specifically, using the Golden-Thompson inequality we have

$$\begin{aligned}&\mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_k}\big) \le \mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}}e^{\eta\mathbf{A}_k}\big)\\ &= \mathbf{Tr}\big(\mathbf{W}_k e^{\eta\mathbf{A}_k}\big) \approx \mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}}\big) + \eta\mathbf{W}_k \bullet \mathbf{A}_k \ .\end{aligned}$$

One can also use convexity to show

$$\mathbf{Tr}\big(e^{c_{k+1}\mathbf{I}+\eta\mathbf{\Sigma}_k}\big) - \mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_k}\big) \le c_{k+1} - c_k \ .$$

Adding these two inequalities, and using the fact that $\mathbf{Tr}\mathbf{W}_k = \mathbf{Tr}\mathbf{W}_{k+1} = 1$, we immediately have $c_k - c_{k+1} \lesssim \eta\mathbf{W}_k \bullet \mathbf{A}_k$. In other words, the gain value $\mathbf{W}_k \bullet \mathbf{A}_k$ at iteration k, up to a factor η, is lower bounded by the decrement of c_k. On the other hand, it is easy to see $c_1 - c_{T+1} \ge \eta\lambda_{\max}(\mathbf{\Sigma}_T) - O(\log d)$ from $c_1 = -\log d$ and the definition of c_{T+1}. Together, we can derive that

$$\textstyle\sum_{k=1}^{T}\mathbf{W}_k \bullet \mathbf{A}_k \gtrsim \lambda_{\max}(\mathbf{\Sigma}_T) \ .$$

In the rest of this section, we perform a thought experiment to "modify" the above MMWU analysis step-by-step. In the end, the intuition of our FTCL shall become clear to the reader.

Thinking Step 1. We wish to choose a random Gaussian vector $u \in \mathbb{R}^d$ and "compress" MMWU to dimension 1 in the direction of u. More specifically, we define $\mathbf{W}_k = \exp(c_k\mathbf{I} + \eta\mathbf{\Sigma}_{k-1})$ but this time c_k is the unique constant such that $\mathbf{Tr}(\mathbf{W}_k uu^\top) = u^\top\mathbf{W}_k u = 1$. In such a case, we *wish* to say that

$$\begin{aligned}&\mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_k}uu^\top\big) = \mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}+\eta\mathbf{A}_k}uu^\top\big)\\ &\overset{(\star)}{\le} \mathbf{Tr}\big(e^{(c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1})/2}uu^\top e^{(c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1})/2}e^{\eta\mathbf{A}_k}\big)\\ &= \mathbf{Tr}\big(\mathbf{W}_k^{1/2}uu^\top\mathbf{W}_k^{1/2}e^{\eta\mathbf{A}_k}\big)\\ &\approx \mathbf{Tr}(\mathbf{W}_k uu^\top) + \eta\mathbf{W}_k^{1/2}uu^\top\mathbf{W}_k^{1/2} \bullet \mathbf{A}_k \ .\end{aligned}$$

If the above inequality were true, then we could define $w_k := \mathbf{W}_k^{1/2}u$ which is a unit vector (because $\mathbf{Tr}(\mathbf{W}_k uu^\top) = 1$) and the gain $w_k^\top\mathbf{A}_k w_k = w_k w_k^\top \bullet \mathbf{A}_k$ would again be proportional to the change of this new potential function $\mathbf{Tr}\big(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}}uu^\top\big)$. This idea almost worked except that inequality $(\star)$ is false due to the non-commutativity of matrices.[7]

Perhaps the most "immediate" idea to fix this issue is to use the randomness of $uu^\top$. Recall that $\mathbf{E}[uu^\top] = \mathbf{I}$ if we choose properly normalize u, and therefore it "seems like" we have $\mathbb{E}[\mathbf{Tr}(\mathbf{W}_k uu^\top)] = \mathbf{Tr}(\mathbf{W}_k)$ and the inequality will go through. Unfortunately, this idea fails for a fundamental reason: the normalization constant c_k depends on

[7]A analogy for this effect can be found in the inequality $\mathbf{Tr}(e^{\mathbf{A}}) \le \mathbf{Tr}(e^{\mathbf{B}})$ for every $\mathbf{A} \preceq \mathbf{B}$. This inequality becomes false when multiplied with $uu^\top$ and in general $e^{\mathbf{A}} \preceq e^{\mathbf{B}}$ is false.

u, so $\mathbf{W}_k$ is *not* independent from the randomness of u.[8]

Thinking Step 2. Since Gaussian vectors are rotationally invariant, we switch wlog to the eigenbasis of $\mathbf{\Sigma}_{k-1}$ so $\mathbf{W}_k$ is a diagonal matrix. We make an important observation:[9]

c_k depends only on $|u_1|, \dots, |u_d|$,

but not on the 2^d possible signs of $u_1, \dots, u_d$.

For this reason, we can fix a diagonal matrix $\mathbf{D}$ and consider all random $uu^\top$ which *agree* with $\mathbf{D}$ on its diagonal,[10] All of such vectors u give the same normalization constant c_k, and it satisfies $\mathbb{E}[uu^\top | \mathbf{D}] = \mathbf{D}$. This implies that we can now study the conditional expected potential change

$$\begin{aligned}\mathbb{E}\left[\mathbf{Tr}\left(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_k} uu^\top\right) - \mathbf{Tr}\left(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}} uu^\top\right) \big| \mathbf{D}\right] \\ = \mathbf{Tr}\left(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_k}\mathbf{D}\right) - \mathbf{Tr}\left(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}}\mathbf{D}\right) \ ,\end{aligned}$$

or if we denote by $\mathbf{B} = c_k\mathbf{I} + \eta\mathbf{\Sigma}_{k-1}$, we want to study the difference $\mathbf{Tr}\left(e^{\mathbf{B}+\eta\mathbf{A}_k}\mathbf{D}\right) - \mathbf{Tr}\left(e^{\mathbf{B}}\mathbf{D}\right)$ only in the special case that $\mathbf{D}$ and $\mathbf{B}$ are *simultaneously diagonalizable*.

Thinking Step 3. A usual way to bound $\mathbf{Tr}\left(e^{\mathbf{B}+\eta\mathbf{A}_k}\mathbf{D}\right) - \mathbf{Tr}\left(e^{\mathbf{B}}\mathbf{D}\right)$ is to define $f(\eta) := \mathbf{Tr}\left(e^{\mathbf{B}+\eta\mathbf{A}_k}\mathbf{D}\right)$ and bound $f(\eta)$ by its Taylor series $f(0) + f'(0)\eta + \frac{1}{2}f''(0)\eta^2 + \cdots$. The zero-order derivative $f(0)$ is $\mathbf{Tr}\left(e^{\mathbf{B}}\mathbf{D}\right)$. The first-order derivative $f'(0) = \mathbf{Tr}(\mathbf{A}_k e^{\mathbf{B}}\mathbf{D}) = e^{\mathbf{B}/2}\mathbf{D}e^{\mathbf{B}/2} \bullet \mathbf{A}_k$ behaves exactly in the way we hope, and this strongly relies on the commutativity between $\mathbf{B}$ and $\mathbf{D}$. Unfortunately, higher-order derivatives $f^{(k)}(0)$ benefit less and less from the commutativity between $\mathbf{B}$ and $\mathbf{D}$ due to the existence of terms such as $\mathbf{A}_k e^{\mathbf{B}}\mathbf{D}e^{\mathbf{B}}\mathbf{A}_k\mathbf{D}$. For this reason, we need to (1) truncate the Taylor series and (2) use different analytic tools. This motivates us to use the following regime that can be viewed as a "low-degree" version of MMWU:

A Quick Detour. In a recent result, the authors of (Allen-Zhu et al., 2015) generalized MMWU to $\ell_{1-1/q}$ regularized strategies. For every $q \geq 2$, they define $\mathbf{X}_k = (c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1})^{-q}$ where c_k is the unique constant such that $c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1} \succ 0$ and $\mathbf{Tr}\mathbf{X}_k = 1$.[11] This is a generalization of MMWU because when $q \approx \log d$, the matrix $\mathbf{X}_k$ behaves nearly the same as $\mathbf{W}_k$; in particular, it gives the same regret bound. The analysis behind this new strategy is to keep track of the potential change in $\mathbf{Tr}\left((c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1})^{-(q-1)}\right)$ as opposed to $\mathbf{Tr}\left(e^{c_k\mathbf{I}+\eta\mathbf{\Sigma}_{k-1}}\right)$, and then use the so-called Lieb-Thirring inequality (see Section 5) to replace the use of Golden-Thompson. (Note that c_k is choosen with respect to q but the potential is with respect to $q-1$.)

Thinking Step 4. Let us now replace MMWU strategies in our Thinking Steps 1,2,3 with $\ell_{1-1/q}$ regularized strategies. Such strategies have two advantages: (1) they help us overcome the issue for higher-order terms in Thinking Step 3, and (2) solving linear systems is *more efficient* than computing matrices exponentials. We shall choose $q = \Theta(\log(dT))$ in the end.

Specifically, we prepare a random vector u and define the normalization constant c_k to be the unique one satisfying $\mathbf{Tr}\left((c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1})^{-q} uu^\top\right) = \mathbf{Tr}(\mathbf{X}_k uu^\top) = 1$. At iteration k, we let the player choose strategy $\mathbf{X}_k^{1/2}u$ which is a unit vector.

If one goes through all the math carefully (using Woodbury formula), this time we are entitled to upper bound the trace difference of the form $\mathbf{Tr}\left((\mathbf{B}+\eta\mathbf{C})^{q-1}\mathbf{D}\right) - \mathbf{Tr}\left(\mathbf{B}^{q-1}\mathbf{D}\right)$ where $\mathbf{D}$ is simultaneously diagonalizable with $\mathbf{B}$ but not $\mathbf{C}$. Similar to Thinking Step 3, we can define $f(\eta) := \mathbf{Tr}\left((\mathbf{B}+\eta\mathbf{C})^{q-1}\mathbf{D}\right)$ and bound this polynomial $f(\eta)$ using its Taylor expansion at point 0. Commutativity between $\mathbf{B}$ and $\mathbf{D}$ helps us compute $f'(0) = (q-1)\mathbf{Tr}(\mathbf{B}^{q-2}\mathbf{C}\mathbf{D})$ but again we cannot bound higher-derivatives directly. Fortunately, this time $f(\eta)$ is a degree $q-1$ polynomial so we can use Markov brothers' inequality to give an upper bound on its higher-order terms. This is the place we lose a few extra polylogarithmic factors in the total regret.

Thinking Step 5. Somehow necessarily, even the second-order derivative $f''(0)$ can depend on terms such as $1/D_{ii}$ where $D_{ii} = |u_i|^2$ is the i-th diagonal entry of $\mathbf{D}$. This quantity, over the Gaussian random choice of u_i, does not have a bounded mean. More generally, the inverse chi-squared distribution with degree t (recall Section 2) has a bounded mean only when $t \geq 3$. For this reason, instead of picking a single random vector $u \in \mathbb{R}^d$, we need to pick three random vectors $u_1, u_2, u_3 \in \mathbb{R}^d$ and replace all the occurrences of $uu^\top$ with $\frac{1}{3}\left(u_1u_1^\top + u_2u_2^\top + u_3u_3^\top\right)$ in the previous thinking steps. As a result, each D_{ii} becomes a chi-squared distribution of degree 3 so the issue goes away. This is why we claimed in the introduction that

we can compress MMWU to dimension 3.

Thinking Step 6. Putting together previous steps, we obtain a FTCL strategy with total regret $O(\sqrt{T}\log^3(dT))$, which is worse than MMWU only by a factor $O(\log^{2.5}(dT))$. We call this method $\texttt{FTCL}^{\mathsf{obl}}$ and include its analysis in Section 6. However, $\texttt{FTCL}^{\mathsf{obl}}$ only works for an oblivious adversary (i.e., when $\mathbf{A}_1, \dots, \mathbf{A}_T$

[8]In fact, c_k can be made almost independent from u if we replace $uu^\top$ with $\mathbf{Q}\mathbf{Q}^\top$ where $\mathbf{Q}$ is a random $d \times m$ matrix for some very large m. That was the main idea behind MMWU-JL.

[9]This is because, $\mathbf{Tr}(e^{c_k\mathbf{I}-\eta\mathbf{\Sigma}_{k-1}} uu^\top) = \sum_{i=1}^d \left(|u_i|^2 \cdot e^{c_k - \eta\lambda_i}\right)$ where λ_i is the i-th eigenvalue of $\mathbf{\Sigma}_{k-1}$.

[10]That is, all random $uu^\top$ such that $\|u_i\|_2^2 = \mathbf{D}_{i,i}$ for each $i \in [d]$. For simplicity we also denote this event as $\mathbf{D}$.

[11]The name "$\ell_{1-1/q}$ strategies" comes from the following fact. Recall MMWU naturally arises as the follow-the-regularized-leader strategy, where the regularizer is the matrix entropy. If the entropy function is replaced with a negative $\ell_{1-1/q}$ norm, the resulting strategy becomes $\mathbf{X}_k$. We encourage interested readers to see the introduction of (Allen-Zhu et al., 2015) for more background, but we shall make this present paper self-contained.

are fixed a priori) and gives an expected regret. To turn it into a robust strategy against *adversarial* $\mathbf{A}_1, \dots, \mathbf{A}_T$, and to make the regret bound work with high confidence, we need to re-sample u_1, u_2, u_3 every iteration. We call this method $\mathtt{FTCL}^{\mathsf{adv}}$. A careful but standard analysis with Azuma inequality helps us reduce $\mathtt{FTCL}^{\mathsf{adv}}$ to $\mathtt{FTCL}^{\mathsf{obl}}$. We state this result in the full version.

Running Time. As long as q is an even integer, the computation of "$(c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1})^{-1}$ applied to a vector" (or in other words, solving linear systems) becomes the bottleneck for each iteration of $\mathtt{FTCL}^{\mathsf{obl}}$ and $\mathtt{FTCL}^{\mathsf{adv}}$. However, as long as $q \geq \Omega(\log(dT))$, we show that the condition number of the matrix $c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1}$ is at most $\eta T = \Theta(T^{1/2})$. Conjugate gradient solves each such linear system in worst-case time $\widetilde{O}(\min\{T^{1/4}, d\} \times \mathsf{nnz}(\mathbf{\Sigma}_{k-1}))$.

Compress to 1-d in Stochastic Online Eigenvector. If the adversary is stochastic, we observe that Oja's algorithm corresponds to a potential function $\mathbf{Tr}\big((\mathbf{I} + \eta\mathbf{A}_k)\cdots(\mathbf{I} + \eta\mathbf{A}_1)uu^\top(\mathbf{I} + \eta\mathbf{A}_1)\cdots(\mathbf{I} + \eta\mathbf{A}_k)\big)$. Because the matrices are drawn from a common distribution, this potential behaves similar to the matrix exponential but compressed to dimension 1, namely $\mathbf{Tr}\big(e^{\eta(\mathbf{A}_1+\cdots+\mathbf{A}_k)}uu^\top\big)$. In fact, just using linearity of expectation carefully, one can both upper and lower bound this potential. We state this result in Section 8 (and it can be proved in one page!)

5 A New Trace Inequality

Prior work on MMWU and its extensions rely heavily on one of the following trace inequalities: the Golden-Thompson inequality

$$\mathbf{Tr}(e^{\mathbf{A}+\eta\mathbf{B}}) \leq \mathbf{Tr}\big(e^{\mathbf{A}}e^{\eta\mathbf{B}}\big)$$

and the Lieb-Thirring inequality

$$\mathbf{Tr}\big((\mathbf{A}+\eta\mathbf{B})^k\big) \leq \mathbf{Tr}\big(\mathbf{A}^{k/2}(\mathbf{I}+\eta\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})^k\mathbf{A}^{k/2}\big) \ .$$

Due to our compression framework in this paper, we need inequalities of type

$$\text{“ } \mathbf{Tr}(e^{\mathbf{A}+\eta\mathbf{B}}\mathbf{D}) \leq \mathbf{Tr}\big(e^{\eta\mathbf{B}}e^{\mathbf{A}/2}\mathbf{D}e^{\mathbf{A}/2}\big) \text{ ”}$$

$$\text{“ } \mathbf{Tr}\big((\mathbf{A} + \eta\mathbf{B})^k\mathbf{D}\big) \leq \mathbf{Tr}\big((\mathbf{I} + \eta\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})^k\mathbf{A}^{k/2}\mathbf{D}\mathbf{A}^{k/2}\big) \text{ ”} \quad (5.1)$$

which look almost like "generalizations" of Golden-Thompson and Lieb-Thirring (by setting $\mathbf{D} = \mathbf{I}$). Unfortunately, such generalizations ***do not hold*** for an arbitrary $\mathbf{D}$. For instance, if the first "generalization" holds for every PSD matrix $\mathbf{D}$ then it would imply " $e^{\mathbf{A}+\eta\mathbf{B}} \preceq e^{\mathbf{A}/2}e^{\eta\mathbf{B}}e^{\mathbf{A}/2}$ " which is a false inequality due to matrix non-commutativity.

In this paper, we show that if $\mathbf{D}$ is *commutative* with $\mathbf{A}$, then the "generalization" (5.1) holds *for the zeroth and first order terms* with respect to η. As for higher order terms, we can control it using Markov brothers' inequality. (Proof see full paper.)

Lemma 5.1. *For every symmetric matrices $\mathbf{A}, \mathbf{B}, \mathbf{D} \in \mathbb{R}^{d\times d}$, every integer $k \geq 1$, every $\eta^* \geq 0$, and every $\eta \in [0, \eta^*/k^2]$, if $\mathbf{A}$ and $\mathbf{D}$ are commutative, then*

$$(\mathbf{A} + \eta\mathbf{B})^k \bullet \mathbf{D} - \mathbf{A}^k \bullet \mathbf{D} \leq k\eta\mathbf{B} \bullet \mathbf{A}^{k-1}\mathbf{D}$$
$$+ \left(\frac{\eta k^2}{\eta^*}\right)^2 \max_{\eta' \in [0,\eta^*]} \Big\{\big|(\mathbf{A} + \eta'\mathbf{B})^k \bullet \mathbf{D} - \mathbf{A}^k \bullet \mathbf{D}\big|\Big\} \ .$$

6 Oblivious Online EV + Expected Regret

In this section we first focus on a simpler oblivious setting. $\mathbf{A}_1, \dots, \mathbf{A}_T$ are T PSD matrices chosen by the adversary *in advance*, and they do not depend on the player's actions in the T iterations. We are interested in upper bounding the total *expected* regret

$$\lambda_{\max}\big(\textstyle\sum_{k=1}^T \mathbf{A}_k\big) - \sum_{k=1}^T \mathbb{E}[w_k^\top \mathbf{A}_k w_k] \ ,$$

where the expectation is over player's random choices $w_k \in \mathbb{R}^d$ (recall $\|w_k\|_2 = 1$).

In the full version, we generalize this result to the full adversarial setting along with high-confidence regret.

Our algorithm $\mathtt{FTCL}^{\mathsf{obl}}$ is presented in Algorithm 1. It is parameterized by an even integer $q \geq 2$ and a learning rate $\eta > 0$. It initializes with a rank-3 Wishart random matrix $\mathbf{U}$. For every $k \in [T+1]$, we denote by $\mathbf{X}_k := \big(c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1}\big)^{-q}$ where $c_k > 0$ is the unique constant s.t.[12]

$$c_k\mathbf{I} - \eta\mathbf{\Sigma}_{k-1} \succ 0 \quad \text{and} \quad \mathbf{Tr}\big(\mathbf{X}_k\mathbf{U}\big) = 1 \ .$$

At iteration $k \in [T]$, the player plays a random unit vector w_k, among the three eigenvectors of $\mathbf{X}_k^{1/2}\mathbf{U}\mathbf{X}_k^{1/2}$. It satisfies $\mathbf{E}[w_k w_k^\top] = \mathbf{X}_k^{1/2}\mathbf{U}\mathbf{X}_k^{1/2}$.

We prove the following theorem for the total regret of $\mathtt{FTCL}^{\mathsf{obl}}(T, q, \eta)$ in the full version of this paper:

Theorem 1. *If we have an oblivious adversary, there exists absolute constant $C > 1$ such that if $q \geq 3\log(2dT)$ and $\eta \in \big[0, \frac{1}{11q^3}\big]$, then $\mathtt{FTCL}^{\mathsf{obl}}(T, q, \eta)$ satisfies*

$$\sum_{k=1}^T \mathbb{E}\left[w_k^\top \mathbf{A}_k w_k\right] \geq \big(1 - C\eta q^5 \log(dT)\big)\lambda_{\max}(\mathbf{\Sigma}_T) - \frac{4}{\eta}$$

Corollary 6.1. *If $q = 3\log(2dT)$ and $\eta = \Theta\big(\frac{\log^{-3}(dT)}{\sqrt{\lambda_{\max}(\mathbf{\Sigma}_T)}}\big)$*

$$\textstyle\sum_{k=1}^T \mathbb{E}\left[w_k^\top \mathbf{A}_k w_k\right] \qquad (\lambda\text{-refined language})$$
$$\geq \lambda_{\max}(\mathbf{\Sigma}_T) - O\Big(\sqrt{\lambda_{\max}(\mathbf{\Sigma}_T)}\log^3(dT)\Big) \ ,$$

or choosing the same q but $\eta = \Theta\big(\frac{\log^{-3}(dT)}{\sqrt{T}}\big)$ we have

$$\textstyle\sum_{k=1}^T \mathbb{E}\left[w_k^\top \mathbf{A}_k w_k\right] \geq \lambda_{\max}(\mathbf{\Sigma}_T) - O\Big(\sqrt{T}\log^3(dT)\Big) \ . \qquad (\text{general language})$$

[12]This c_k is unique because $\mathbf{Tr}\big(\mathbf{X}_k\mathbf{U}\big)$ is a strictly decreasing function for $c_k > \eta\lambda_{\max}(\mathbf{\Sigma}_{k-1})$.

Algorithm 1 $\texttt{FTCL}^{\mathsf{obl}}(T, q, \eta)$

```
Input: T, number of iterations; q ≥ 2, an even integer,      ◇ theory-predicted choice $q = \Theta(\log(dT))$
       η, the learning rate.                                  ◇ theory-predicted choice $\eta = \log^{-3}(dT)/\sqrt{\lambda_{\max}(\mathbf{\Sigma}_T)}$
1: Choose $u_1, u_2, u_3 \in \mathbb{R}^d$ where the $3d$ coordinates are i.i.d. drawn from $\mathcal{N}(0,1)$.
2: $\mathbf{U} \leftarrow \frac{1}{3}\big(u_1 u_1^\top + u_2 u_2^\top + u_3 u_3^\top\big)$.
3: for $k \leftarrow 1$ to $T$ do
4:     $\mathbf{\Sigma}_{k-1} \leftarrow \sum_{i=1}^{k-1} \mathbf{A}_i$.
5:     Denote by $\mathbf{X}_k \leftarrow \big(c_k \mathbf{I} - \eta \mathbf{\Sigma}_{k-1}\big)^{-q}$ where $c_k$ is the constant s.t. $c_k \mathbf{I} - \eta \mathbf{\Sigma}_{k-1} \succ 0$ and $\mathbf{Tr}\big(\mathbf{X}_k \mathbf{U}\big) = 1$ .
6:     Compute $\mathbf{X}_k^{1/2} \mathbf{U} \mathbf{X}_k^{1/2} = \sum_{j=1}^3 p_j \cdot y_j y_j^\top$ where $y_1, y_2, y_3$ are orthogonal unit vectors in $\mathbb{R}^d$.
7:     Choose $w_k \leftarrow y_j$ with probability $p_j$.     ◇ it satisfies $p_1, p_2, p_3 \geq 0$ and $p_1 + p_2 + p_3 = 1$.
8:     Play strategy $w_k$ and receive matrix $\mathbf{A}_k$.
9: end for
```

7 Missing Theorems

In the full version of this paper, we state Theorem 2, a similar result of Theorem 1 but (1) with a high-confidence regret bound and (2) for the adversarial setting: $\mathbf{A}_k$ may depend on the player's strategies $w_1, \dots, w_{k-1}$. In the full version, we also provide Theorem 3, which addresses the worst-case complexity for implementing the matrix inversion steps in $\texttt{FTCL}^{\mathsf{obl}}$.

8 Stochastic Online Eigenvector

Consider the special case when the matrices $\mathbf{A}_1, \dots, \mathbf{A}_T$ are generated i.i.d. from a common distribution whose expectation equals $\mathbf{B}$. This is known as the stochastic online eigenvector problem, and we wish to minimize the regret

$$\textstyle\sum_{k=1}^T w_k^\top \mathbf{A}_k w_k - T \cdot \lambda_{\max}(\mathbf{B}) \ .$$

We revisit Oja's algorithm: beginning with a random Gaussian vector $u \in \mathbb{R}^d$, at each iteration k, let w_k be $(\mathbf{I} + \eta \mathbf{A}_{k-1}) \cdots (\mathbf{I} + \eta \mathbf{A}_1) u$ after normalization. It is clear that w_k can be computed from w_{k-1} in time $\mathsf{nnz}(\mathbf{A})$.

In the full version, we prove the next theorem in one page:

Theorem 4. *There exists $C > 1$ such that, for every $p \in (0,1)$, if $\eta \in \big[0, \sqrt{p/(60 T \lambda_{\max}(\mathbf{B}))}\big]$ in Oja's algorithm, we have with probability at least $1-p$:*

$$\textstyle\sum_{k=1}^T w_k^\top \mathbf{A}_k w_k \geq (1-2\eta) T \cdot \lambda_{\max}(\mathbf{B}) - C \cdot \frac{\log(d + \log(1/p))}{\eta} \ .$$

Corollary 8.1. *Choosing $\eta = \sqrt{p}/\sqrt{60 T \lambda_{\max}(\mathbf{B})}$, we have with prob. $\geq 1-p$:*

$$\textstyle\sum_{k=1}^T w_k^\top \mathbf{A}_k w_k \geq T \cdot \lambda_{\max}(\mathbf{B}) - O\Big(\frac{\sqrt{T \cdot \lambda_{\max}(\mathbf{B})}}{\sqrt{p}} \cdot \log(d + \log(1/p))\Big) \ .$$

(λ-refined language)

Choosing $\eta = \sqrt{p}/\sqrt{60T}$, we have with prob. $\geq 1-p$:

$$\textstyle\sum_{k=1}^T w_k^\top \mathbf{A}_k w_k \geq T \cdot \lambda_{\max}(\mathbf{B}) - O\Big(\frac{\sqrt{T}}{\sqrt{p}} \cdot \log(d + \log(1/p))\Big) \ .$$

(general language)

The proof of Theorem 4 uses a potential function analysis which is similar to the matrix exponential potential used in MMWU, but compressed to dimension 1.

9 Conclusions

We give a new learning algorithm FTCL for the online eigenvector problem. It matches the optimum regret obtained by MMWU, but runs *much faster*. It matches the fast per-iteration running time of FTPL, but has a *much smaller regret*. In the stochastic setting, our side result on Oja's algorithm also outperforms previous results. We believe our novel idea of "follow the compressed leader" may find other applications in the future.

Acknowledgement

We thank Yin Tat Lee for discussing the problem regarding how to compress MMWU to constant dimension in 2015. We thank Elad Hazan for suggesting us the problem and for several insightful discussions. We thank Dan Garber and Tengyu Ma for clarifying some results of prior work (Garber et al., 2015).

References

Abernethy, Jacob, Lee, Chansoo, Sinha, Abhinav, and Tewari, Ambuj. Online linear optimization via smoothing. In *COLT*, pp. 807–823, 2014.

Abernethy, Jacob, Lee, Chansoo, and Tewari, Ambuj. Spectral smoothing via random matrix perturbations. *ArXiv e-prints*, abs/1507.03032, 2015.

Allen-Zhu, Zeyuan and Li, Yuanzhi. LazySVD: Even Faster SVD Decomposition Yet Without Agonizing Pain. In *NIPS*, 2016a.

Allen-Zhu, Zeyuan and Li, Yuanzhi. First Efficient Convergence for Streaming k-PCA: a Global, Gap-Free, and Near-Optimal Rate. *ArXiv e-prints*, abs/1607.07837, July 2016b.

Allen-Zhu, Zeyuan and Li, Yuanzhi. Doubly Accelerated Methods for Faster CCA and Generalized Eigendecomposition. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, 2017.

Allen-Zhu, Zeyuan and Yuan, Yang. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*, 2016.

Allen-Zhu, Zeyuan, Liao, Zhenyu, and Orecchia, Lorenzo. Spectral Sparsification and Regret Minimization Beyond Multiplicative Updates. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, STOC '15, 2015.

Allen-Zhu, Zeyuan, Lee, Yin Tat, and Orecchia, Lorenzo. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, 2016.

Arora, Sanjeev and Kale, Satyen. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*, pp. 227, New York, New York, USA, 2007. ACM Press. ISBN 9781595936318. doi: 10.1145/1250790.1250823.

Arora, Sanjeev, Hazan, Elad, and Kale, Satyen. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8:121–164, 2012. doi: 10.4086/toc.2012.v008a006.

Boutsidis, Christos, Garber, Dan, Karnin, Zohar, and Liberty, Edo. Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 887–901. SIAM, 2015.

Bunch, James R and Hopcroft, John E. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

Cesa-Bianchi, Nicolo and Lugosi, Gabor. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, 2006. ISBN 9780511546921. doi: 10.1017/CBO9780511546921.

Dwork, Cynthia, Talwar, Kunal, Thakurta, Abhradeep, and Zhang, Li. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *STOC*, pp. 11–20. ACM, 2014.

Frostig, Roy, Ge, Rong, Kakade, Sham M., and Sidford, Aaron. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, volume 37, pp. 1–28, 2015. URL `http://arxiv.org/abs/1506.07512`.

Garber, Dan and Hazan, Elad. Fast and simple PCA via convex optimization. *ArXiv e-prints*, September 2015.

Garber, Dan, Hazan, Elad, and Ma, Tengyu. Online learning of eigenvectors. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 560–568, 2015.

Hardt, Moritz and Price, Eric. The noisy power method: A meta algorithm with applications. In *NIPS*, pp. 2861–2869, 2014.

Jain, Rahul, Ji, Zhengfeng, Upadhyay, Sarvagya, and Watrous, John. QIP = PSPACE. *Journal of the ACM (JACM)*, 58(6):30, 2011.

Karnin, Zohar and Liberty, Edo. Online pca with spectral bounds. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, pp. 505–509, 2015.

Kotłowski, Wojciech and Warmuth, Manfred K. Pca with gaussian perturbations. *ArXiv e-prints*, abs/1506.04855, 2015.

Lee, Yin Tat and Sun, He. Constructing linear-sized spectral sparsification in almost-linear time. In *FOCS*, pp. 250–269. IEEE, 2015.

Lin, Hongzhou, Mairal, Julien, and Harchaoui, Zaid. A Universal Catalyst for First-Order Optimization. In *NIPS*, 2015. URL `http://arxiv.org/pdf/1506.02186v1.pdf`.

Nesterov, Yurii. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004. ISBN 1402075537.

Nie, Jiazhong, Kotłowski, Wojciech, and Warmuth, Manfred K. Online pca with optimal regrets. In *International Conference on Algorithmic Learning Theory*, pp. 98–112. Springer, 2013.

Orecchia, Lorenzo. *Fast Approximation Algorithms for Graph Partitioning using Spectral and Semidefinite-Programming Techniques*. PhD thesis, EECS Department, University of California, Berkeley, May 2011.

Orecchia, Lorenzo, Sachdeva, Sushant, and Vishnoi, Nisheeth K. Approximating the exponential, the lanczos method and an $\widetilde{O}(m)$-time spectral algorithm for balanced separator. In *STOC '12*. ACM Press, November 2012.

Pan, Victor Y and Chen, Zhao Q. The complexity of the matrix eigenproblem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pp. 507–516. ACM, 1999.

Peng, Richard and Tangwongsan, Kanat. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. In *Proceedinbgs of the 24th ACM symposium on Parallelism in algorithms and architectures - SPAA '12*, pp. 101, New York, New York, USA, January 2012.

Shalev-Shwartz, Shai. SDCA without Duality, Regularization, and Individual Convexity. In *ICML*, 2016.

Shamir, Ohad. Convergence of stochastic gradient descent for pca. In *ICML*, 2016.

Shewchuk, Jonathan Richard. An introduction to the conjugate gradient method without the agonizing pain, 1994.

Wendel, J. G. Note on the gamma function. *The American Mathematical Monthly*, 55(9):563–564, 1948.