Online Learning of Eigenvectors

Dan Garber DANGAR@TX.TECHNION.AC.IL
Technion - Israel Institute of Technology

Elad Hazan EHAZAN@CS.PRINCETON.EDU
Princeton University

Tengyu Ma TENGYUL@CS.PRINCETON.EDU
Princeton University

Abstract

Computing the leading eigenvector of a symmetric real matrix is a fundamental primitive of numerical linear algebra with numerous applications. We consider a natural online extension of the leading eigenvector problem: a sequence of matrices is presented and the goal is to predict for each matrix a unit vector, with the overall goal of competing with the leading eigenvector of the cumulative matrix. Existing regret-minimization algorithms for this problem either require to compute an *eigen decompostion* every iteration, or suffer from a large dependency of the regret bound on the dimension. In both cases the algorithms are not practical for large scale applications.

In this paper we present new algorithms that avoid both issues. On one hand they do not require any expensive matrix decompositions and on the other, they guarantee regret rates with a mild dependence on the dimension at most. In contrast to previous algorithms, our algorithms also admit implementations that enable to leverage sparsity in the data to further reduce computation. We extend our results to also handle non-symmetric matrices.

1. Introduction

Computing the leading eigenvector of a symmetric real matrix is one of the most important problems in numerical linear algebra and an important primitive in many algorithms. Perhaps the best known application of this problem is the *Principal Component Analysis* problem, in which, roughly speaking, given a set of high-dimensional vectors, the problem is to find a low-dimensional subspace such that the projection of the vectors onto this low-dimensional subspace is close on average to the original vectors. It is well known that the optimal solution to this problem is to project the high-dimensional vectors over the several leading eigenvectors of the covariance matrix of the data. In this paper we consider an *online learning* problem that is a natural extension of the leading eigenvector problem. A decision maker observes a sequence of matrices. Before a new matrix is revealed, the decision maker must commit to a unit vector. Once the matrix is revealed the decision maker gains the quadratic product of the selected unit vector with the revealed matrix, and his overall goal is to maximize the total reward. As standard in such settings, the performance of the decision maker is measured via the regret which is given by the difference between the total reward of the best fixed unit vector in hindsight and the total reward of the decision maker. Indeed the best fixed unit vector in hindsight is simply given by the leading eigenvector of the sum of revealed matrices and the associated total reward is the corresponding leading eigenvalue. This problem captures as a special case the problem known as *Online Principal Component Analysis* that was studied in (Warmuth & Kuzmin, 2006a), (Warmuth & Kuzmin, 2006b), (Nie et al., 2013), in case of a single principal component.

From an optimization theory point of view, the offline leading eigenvector problem is a non-convex quadratic optimization problem, since w.l.o.g. it requires to maximize a convex function over a non convex set, i.e. the unit sphere. However, it could be solved to high precision via eigendecomposition which takes $O(n^3)$ time where n is the size of the matrix, or via iterative approximation algorithms such as the *Power* or *Lanczos* algorithms whose running time for well-conditioned matrices is roughly $O(\text{nnz})$ where nnz denotes the number of non-zeros entries in the input matrix.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

When considering the online problem, three different approaches come to mind which we now detail.

The Convexification Approach The first approach is to convexify the problem by *lifiting* the decision variable from a unit vector to a matrix, more specifically a positive semidefinite matrix with unit trace. This approach corresponds to the problem of *Online Linear Optimization* over the *Spectrahedron* [1]. This is also the approach taken in previous works on Online PCA (Warmuth & Kuzmin, 2006a), (Warmuth & Kuzmin, 2006b), (Nie et al., 2013). While this approach leads to theoretically efficient algorithms with nearly optimal regret bounds, such as the *Matrix Multiplicative Weights* algorithm (Tsuda et al., 2005; Arora & Kale, 2007), their major drawback is that they require super-linear computation per iteration, i.e. they require to compute a full eigendcomposition which amounts to $O(n^3)$ arithmetic operations per iteration . The latter is true even if the support of the sequence of matrices is sparse.

The Oracle-based Approach A natural approach is to try to reduce the online problem to the offline one. That is, assume that we are given an oracle for the offline problem that given a query matrix returns its leading eigenvector, and derive an online algorithm that is based on making queries to the oracle. Such a reduction is possible either via the *Follow the Perturbed Leader* meta-algorithm (FPL) (Kalai & Vempala, 2005) or by the recent *Online Frank-Wolfe* algorithm presented in (Hazan & Kale, 2012). Both of these algorithms require on each iteration of the online problem to make a single query to the eigenvector oracle which could be implemented using the Power or Lancsoz algorithms mentioned above with complexity of roughly $O(\text{nnz})$ arithmetic operations. While the per-iteration complexity of these methods is potentially much more favorable than the convexification approach and despite the fact that the regret bound guaranteed by FPL is optimal in terms of the length of the sequence, it comes with the price of a large dependence on the dimension. Indeed when designing online learning algorithms, usually the primary goal is optimal dependence on the sequence length, however favorable dependence on the dimension is crucial in order for the proposed method to be of any practical significance.

The Iterative Approach A third approach is to design online algorithms that directly tackle the non-convex optimization problem, i.e. online analogues of iterative algorithms for the offline problem such as the Power Method with an update step that roughly amounts to computing a single matrix-vector product, which is much more efficient than both previous approaches. Such an approach is reminiscent of the *Online Gradient Decent* method presented in (Zinkevich, 2003) which is an online analogue of the gradient descent method for offline convex optimization. For the specific problem of Stochastic Principal Component Analysis, such algorithms with provable guarantees exist (Balsubramani et al., 2013), (Shamir, 2014), however we are not aware of any such method for the online setting considered here.

[1] formally defined as $\{X \in \mathbb{R}^{n\times n} \mid X \succeq 0, \text{Tr}(X) = 1\}$.

Our interest in this work is to study algorithms for the online eigenvector problem that may be of use for large scale instances. Towards this end we part from the convexification approach that was the main approach studied in previous related problems and requires super-linear computations, and focus on the oracle-based and iterative approaches which allow for more efficient implementations and may leverage sparsity in the data.

1.1. Our Results

Our main result is an online algorithm that takes the so called oracle approach and is based on the *Follow the Perturbed Leader* meta-algorithm. The algorithm requires on each iteration to perform only a single call to an offline eigenvector oracle and attains near-optimal regret in terms of the sequence length. In contrast to previous such algorithms, the dependence of the obtained regret bound on the dimension is much more favorable. Moreover, as opposed to previous oracle-based approaches, our algorithm admits an implementation that may leverage sparsity in the data to further reduce computation. On the technical side, our algorithm is based on a novel analysis of FPL. While previous approaches to analyzing the FPL algorithm are geometric in nature, which seems to inevitably introduce a large dependence on the dimension, our approach exploits the specific structure of the problem at hand and is algebraic in nature. More precisely, we study the spectrum of symmetric real matrices under Gaussian perturbations and apply tools from *matrix perturbation theory* to derive the regret bound.

We also consider a somewhat easier stochastic setting, in which we assume that the sequence of matrices is sampled from a fixed and unknown distribution. We present an algorithm that takes the so called iterative approach and is analogues to the Power algorithm for the offline setting, i.e. it computes a single matrix-vector product on each iteration. The regret of the algorithm is nearly optimal in terms of the sequence length and depends on the dimension only through a logarithmic factor. The analysis of the algorithm is especially accessible and requires only a black-box application of the offline Power method and a, by now standard, matrix concentration inequality.

A comparison of our results to previous related work is detailed in Table 1.

Method	Regret bound	Iteration complexity
Matrix Mul. Weights (Tsuda et al., 2005; Arora & Kale, 2007)	$\sqrt{T}$	n^3 (SVD)
Follow the Perturbed Leader (Kalai & Vempala, 2005) (see Subsection 4.1)	$n^{5/4}\sqrt{T}$	EV
Online Frank-Wolfe (Hazan & Kale, 2012)	$n^{1/2}T^{3/4}$	EV
This paper, online setting (see Section 5)	$\sqrt{nT}$	EV
This paper, stochastic setting (see Section 3)	$\sqrt{T}$	nnz

Table 1. Comparison between different algorithms for the online eigenvector problem. We denote by EV the computation of the leading eigenvector of a given matrix. We omit the dependence on constants and logarithmic factors in n, T.

A performance measure that seems natural for comparing between online algorithms with different regret bounds and iteration complexity is the worst case overall time complexity to achieve ϵ average (expected) regret. This measure is important for instance when considering the application of online learning algorithms to saddle-point optimization problems (Grigoriadis & Khachiyan, 1995; Clarkson et al., 2012; Garber & Hazan, 2011). The overall complexity required for the Matrix Mul. Weights method to achieve ϵ average regret is $\tilde{O}(n^3\epsilon^{-2})$. Our online algorithm on the other hand admits an implementation with overall running time $\tilde{O}(n^{3/2}\epsilon^{-7/2}\text{nnz})$, where nnz denotes the joint-sparsity of observed matrices (see subsection 5.1 for details). Hence in case $\epsilon^{-3/2}\text{nnz} = \tilde{O}(n^{3/2})$, it is overall faster.

Finally, we extend our results to handle non-symmetric matrices as well.

2. Notation and Problem Setting

We denote by $\mathbb{S}_n$ the linear space of all $n \times n$ real symmetric matrices. We denote by $\mathcal{S}$ the Euclidean sphere in $\mathbb{R}^n$, that is $\mathcal{S} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$. Given a matrix $A \in \mathbb{S}_n$ we denote its eigenvalues by $\lambda_1(A) \geq \lambda_2(A) \geq ...\lambda_n(A)$. We also refer to $\lambda_1(A)$ as $\lambda_{\max}$ which is given by $\lambda_{\max} = \max_{x\in\mathcal{S}} x^\top Ax$. We denote by $\delta(A)$ the *eigengap* of the matrix A which is given by $\delta(A) = \lambda_1(A) - \lambda_2(A)$. Unless specified else, given a vector $x \in \mathbb{R}^n$ we denote by $\|x\|$ its standard Euclidean norm and given a matrix $A \in \mathbb{S}_n$ we denote by $\|A\|$ its spectral norm. We also denote by $\|A\|_F$ and $\|A\|_*$ the Frobenius and nuclear norms of A respectively. Recall that $\|A\|, \|A\|_*$ are dual norms and thus according to Holder's inequality, it holds for any two matrices $A, B \in \mathbb{S}_n$ that $A \bullet B \leq \|A\| \cdot \|B\|_*$ where $\bullet$ denotes the standard inner product for matrices, that is $A \bullet B = \sum_{i,j} A_{i,j} \cdot B_{i,j}$.

In this work we consider the following repeated game: an adversary chooses a sequence of matrices $A_1, A_2, ..., A_T \in \mathbb{S}_n$. Then, for T rounds, the player is required on each round $t \in [T]$ to choose a vector $x_t \in \mathcal{S}$. After making his choice, the matrix A_t is revealed and the player gains the profit $x_t^\top A_t x_t$. Such an adversary is referred to in the literature as *oblivious* since he chooses the sequence of matrices without any knowledge of the actual actions of the player. A stronger type of adversary, known as *adaptive adversary*, need not commit in advance to the entire sequence of matrices, but may choose on time t the matrix A_t to depend on the entire history of the game, that is on $A_1, ..., A_{t-1}$ and $x_1, ..., x_{t-1}$. Throughout the paper (especially in Sections 4, 5) we consider only an oblivious adversary. In the full version of the paper we detail how our results could be easily extended to also handle an adaptive adversary.

We measure the overall performance of the player according to the *regret* which is given by.

$$\begin{aligned}\text{regret}_T &= \max_{x\in\mathcal{S}} \sum_{t=1}^{T} x^\top A_t x - \sum_{t=1}^{T} x_t^\top A_t x_t \\ &= \lambda_{\max}\left(\sum_{t=1}^{T} A_t\right) - \sum_{t=1}^{T} x_t^\top A_t x_t.\end{aligned}$$

In case the decision maker uses randomization to choose is actions it also makes sense to consider the expected regret which is given by

$$\mathbb{E}[\text{regret}_T] = \max_{x\in\mathcal{S}} \sum_{t=1}^{T} x^\top A_t x - \mathbb{E}[\sum_{t=1}^{T} x_t^\top A_t x_t],$$

where the expectation is taken over the randomness introduced by the decision maker.

We assume without losing generality that $\|A_t\| \leq 1$ and that A_t is positive definite (note that adding a multiplicity of the identity matrix to A_t does not change the regret).

2.1. The Asymmetric Case

It makes sense to also consider the asymmetric case in which the input matrices $A_1, ..., A_T$ are not necessarily symmetric but are $m \times n$ real matrices for fixed m, n. In this case a prediction is a rank-one matrix $uv^\top$ where $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ and both are unit vectors. In this case the

regret is given by

$$\text{regret}_T = \max_{\substack{u \in \mathbb{R}^m,\ \|u\|=1 \\ v \in \mathbb{R}^n,\ \|v\|=1}} \sum_{t=1}^{T} u^\top A_t v - \sum_{t=1}^{T} u_t^\top A_t v_t$$
$$= \sigma_{\max}\left(\sum_{t=1}^{T} A_t\right) - \sum_{t=1}^{T} u_t^\top A_t v_t,$$

where $\sigma_{\max}(\cdot)$ denotes the largest singular value.

We now show how given a low-regret algorithm for the symmetric problem we can use it to achieve low-regret on the asymmetric problem via a randomized conversion. The algorithm is given below. The following lemma bounds the

Algorithm 1 Asymmetric to Symmetric Conversion Algorithm

1: Input: Algorithm $\mathcal{A}$ for the online eigenvector problem in dimension $m \times n$.
2: **for** $t = 1, ..., T$ **do**
3: Receive prediction $x_t \in \mathbb{R}^{m+n}$ from $\mathcal{A}$.
4: Decompose x_t into $x_t = (\tilde{u}_t, \tilde{v}_t)$ for $\tilde{u}_t \in \mathbb{R}^m, \tilde{v}_t \in \mathbb{R}^n$.
5: With probability $2\|\tilde{u}_t\|\|\tilde{v}_t\|$ set $(u_t, v_t) = \left(\frac{\tilde{u}_t}{\|\tilde{u}_t\|}, \frac{\tilde{v}_t}{\|\tilde{v}_t\|}\right)$. and with remaining probability set $(u_t, v_t) = (u, v)$, for uniformly chosen unit vectors $u \in \mathbb{R}^m, v \in \mathbb{R}^n$.
6: Observe A_t.
7: Feed the matrix $\tilde{A}_t = \begin{pmatrix} 0_{m\times m} & A_t \\ A_t^\top & 0_{n\times n} \end{pmatrix}$ to $\mathcal{A}$.
8: **end for**

regret of Algorithm 1 in terms of the regret of an algorithm for the symmetric problem [2]. The proof is given in the appendix.

Lemma 1. *Assume that for all $t \in [T]$ it holds that $\|A_t\| \le 1$. Then it holds for all $t \in [T]$ that $\tilde{A}_t$ is symmetric, $\|\tilde{A}_t\| \le 1$ and*

$$\mathbb{E}[\sigma_{\max}\left(\sum_{t=1}^{T} A_t\right) - \sum_{t=1}^{T} u_t^\top A_t v_t] =$$
$$\lambda_{\max}\left(\sum_{t=1}^{T} \tilde{A}_t\right) - \sum_{t=1}^{T} x_t^\top \tilde{A}_t x_t,$$

where the expectation is taken over the randomness in choosing u_t, v_t.

[2] Note that the matrix $\tilde{A}_t$ defined in the algorithm is not positive definite as we assumed in the eigenvector problem, however this could be easily fixed by adding a multiplicity of the identity matrix and scaling accordingly to keep the unit upper bound on the spectral norm.

3. The Stochastic Setting

In this section we consider a stochastic setting which is somewhat easier than the online adversarial setting. In the stochastic setting we assume that all matrices $A_1, A_2, ..., A_T$ are sampled i.i.d. from a fixed but unknown distribution $\mathcal{D}$ over matrices in $\mathbb{S}_n$ with spectral norm bounded by one and w.l.o.g. we assume that these matrices are positive definite . We denote the distribution mean by $A = \mathbb{E}_{M\sim\mathcal{D}}[M]$.

Our algorithm titled *Epoch Power Method* for the stochastic setting is given below. It works by dividing the sequence into disjoint epochs, each of length ℓ which is a parameter that will be determined in the analysis. The algorithm predicts using a single unit vector throughout each epoch, while applying Power method update steps in order to compute the prediction for the following epoch.

We refer to an epoch by τ and denote by $x_{(\tau)}$ the point played throughout epoch τ. We also denote $\bar{A}_{\to(\tau)} = \frac{1}{\tau\ell}\sum_{t=1}^{\tau\ell} A_t$, that is the empirical mean of all matrices observed until the end of epoch τ.

In this section we prove the following theorem.

Theorem 1. *Let $x_1, x_2, ..., x_T$ denote the unit vectors played by Algorithm 2 throughout rounds $1, 2, ..., T$. The following guarantees hold.*

1. *Given $\delta > 0$, choosing block length $\ell = \lceil \frac{1}{4}\sqrt{T \log \frac{2nT}{\delta}} \rceil$ guarantees that with probability at least $1 - \delta$*
$$\sum_{t=1}^{T} \left(\lambda_{\max}(A) - x_t^\top A x_t\right) = O\left(\sqrt{T \log \frac{nT}{\delta}}\right).$$
2. *Choosing block length $\ell = \lceil \frac{1}{4}\sqrt{2T \log 2nT} \rceil$ guarantees that*
$$\mathbb{E}[\lambda_{\max}\left(\sum_{t=1}^{T} A_t\right) - \sum_{t=1}^{T} x_t^\top A_t x_t] = O\left(\sqrt{T \log nT}\right).$$

We note that both results in the theorem are optimal up to log factors.

In what follows we always refer to $x_{(\tau)}$ as the final value of this vector, that is, its value at the end of epoch $\tau - 1$.

In order to prove Theorem 1 we need the following two lemmas.

The following lemma is a straightforward application of a Bernstein concentration inequality for symmetric matrices (see (Tropp, 2012), theorem 1.4).

Lemma 2. *For any block τ and $\epsilon > 0$ it holds that*

$$\Pr\left(\|A - \bar{A}_{\to(\tau)}\| \ge \epsilon\right) \le n \exp\left(\frac{-\epsilon^2 \min\{\tau\ell, T\}}{16}\right).$$

Algorithm 2 Epoch Power Method

Input: block length parameter ℓ.
Let v be a unit vector chosen uniformly at random from the sphere.
for $t = 1, ..., \ell$ **do**
  Play arbitrarily & observe A_t.
end for
$x_{(2)} \leftarrow v$.
for $\tau = 2...\lceil T/\ell \rceil$ **do**
  $x_{(\tau+1)} \leftarrow v$.
  Let $\bar{A}_{\to(\tau-1)} = \frac{1}{(\tau-1)\ell} \sum_{t=1}^{(\tau-1)\ell} A_t$.
  for $t = (\tau-1)\ell + 1... \min\{\tau\ell, T\}$ **do**
    Play $x_{(\tau)}$ & observe A_t.
    $x_{(\tau+1)} \leftarrow \bar{A}_{\to(\tau-1)} x_{(\tau+1)} / \|\bar{A}_{\to(\tau-1)} x_{(\tau+1)}\|$.
  end for
end for

The following lemma is based on an analysis of the Power Method for computing the leading eigenvector of a positive definite matrix and gives a guarantee on the quality of the vectors $x_{(\tau)}$. For details see Theorem 4.1 in (Kuczyński & Woźniakowski, 1992).

Lemma 3. *For any $\lfloor \frac{T}{\ell} \rfloor \geq \tau \geq 2$ and $\epsilon > 0$, it holds with probability at least $1 - n\exp(-\ell\epsilon)$ that*

$$x_{(\tau+1)}^\top \bar{A}_{\to(\tau-1)} x_{(\tau+1)} \geq (1-\epsilon)\lambda_{\max}(\bar{A}_{\to(\tau-1)}).$$

We can now prove Theorem 1.

Proof. We first prove part 1 of the theorem, part 2 follows as a corollary.

Fix a block number $\tau \geq 3$ and an error tolerance ϵ_τ. Using Lemmas 2, 3 we have that with probability at least $1 - n\exp(-\ell\epsilon_\tau) - n\exp\left(\frac{\epsilon_\tau^2(\tau-2)\ell}{16}\right)$ that

$$\begin{aligned} &x_{(\tau)} A x_{(\tau)} \geq x_{(\tau)} \bar{A}_{\to(\tau-2)} x_{(\tau)} - \epsilon_\tau \\ &\geq \lambda_{\max}(\bar{A}_{\to(\tau-2)}) - 2\epsilon_\tau \geq \lambda_{\max}(A) - 3\epsilon_\tau, \end{aligned}$$

where the first and last inequalities follow from Lemma 2 and the second inequity follows from Lemma 3 and the observation that $\lambda_{\max}(\bar{A}_{\to(\tau-2)}) \leq 1$.

Setting $\epsilon_\tau = 4\sqrt{\frac{\log \frac{2nT}{\delta}}{(\tau-2)\ell}}$ we have that with probability at least

$$\begin{aligned} &1 - n\exp\left(-4\sqrt{\frac{\ell \log \frac{2nT}{\delta}}{(\tau-2)}}\right) - \frac{\delta}{2T} \\ &> 1 - n\exp\left(-4\sqrt{\frac{\ell^2 \log \frac{2nT}{\delta}}{T}}\right) - \frac{\delta}{2T} \end{aligned}$$

it holds that

$$x_{(\tau)} A x_{(\tau)} \geq \lambda_{\max}(A) - 12\sqrt{\frac{\log \frac{2nT}{\delta}}{(\tau-2)\ell}}.$$

Thus setting the block length to $\ell = \lceil \frac{1}{4}\sqrt{T \log \frac{2nT}{\delta}} \rceil$ we have that with probability at least $1 - \frac{\delta}{T}$ it holds that

$$x_{(\tau)} A x_{(\tau)} \geq \lambda_{\max}(A) - 24\sqrt{\frac{\sqrt{\log \frac{2nT}{\delta}}}{(\tau-2)\sqrt{T}}}.$$

Summing over $\tau \geq 3$ we have that with probability at least $1 - \delta$,

$$\begin{aligned} &\sum_{\tau=3}^{\lceil T/\ell \rceil} \left(x_{(\tau)}^\top A x_{(\tau)} - \lambda_{\max}(A) \right) \leq \\ &24 \left(\frac{\log \frac{2nT}{\delta}}{T} \right)^{1/4} \sum_{\tau=3}^{\lceil T/\ell \rceil} \frac{1}{\sqrt{\tau-2}} < \\ &24 \left(\frac{\log \frac{2nT}{\delta}}{T} \right)^{1/4} \int_1^{T/\ell} \frac{1}{\sqrt{\tau-1}} d\tau < \\ &48 \left(\frac{\log \frac{2nT}{\delta}}{T} \right)^{1/4} \sqrt{\frac{T}{\ell}} = 48 \left(\frac{T \log \frac{2nT}{\delta}}{\ell^2} \right)^{1/4} = 24. \end{aligned}$$

Since each block accounts for ℓ iterations and in worst case the algorithm suffers loss of 1 on all first 2ℓ iterations we have that with probability at least $1 - \delta$

$$\sum_{t=1}^{T} \left(\lambda_{\max}(A) - x_t^\top A x_t \right) \leq 26\ell = O\left(\sqrt{T \log \frac{2nT}{\delta}} \right). \tag{1}$$

We now prove part 2 of the theorem.

Since on any time t, x_t is independent of A_t we have that

$$\begin{aligned} &\mathbb{E}[\max_{x \in \mathcal{S}} \sum_{t=1}^{T} x^\top A_t x - \sum_{t=1}^{T} x_t^\top A_t x_t] \leq \\ &\mathbb{E}[\max_{x \in \mathcal{S}} x^\top (T \cdot A) x + \|\sum_{t=1}^{T} (A_t - A)\| - \sum_{t=1}^{T} x_t^\top A x_t] = \\ &\mathbb{E}[T \cdot \lambda_{\max}(A) - \sum_{t=1}^{T} x_t^\top A x_t + \|\sum_{t=1}^{T} (A_t - A)\|]. \end{aligned} \tag{2}$$

Let us denote by τ_{end} the index of the last block. Note that $\|\sum_{t=1}^{T} A_t - A\| = T \cdot \|\bar{A}_{\to(\tau_{end})} - A\|$. By applying Lemma 2 we have that with probability at least $1 - \delta$ it holds that

$$\frac{1}{T} \|\sum_{t=1}^{T} A_t - A\| = \|\bar{A}_{\to(\tau_{end})} - A\| \leq 4\sqrt{\frac{\ln \frac{n}{\delta}}{T}}. \tag{3}$$

Plugging Eq. (1) and Eq. (3) into Eq. (2) we have that

$$\mathbb{E}[\max_{x\in\mathcal{S}}\sum_{t=1}^{T}x^\top A_t x-\sum_{t=1}^{T}x_t^\top A_t x_t]\leq$$
$$(1-2\delta)\cdot(4+13/2)\sqrt{T\log\frac{2nT}{\delta}}+2\delta\cdot 2T.$$

Thus setting $\delta = T^{-1}$, which corresponds to setting the block length to $\ell = \lceil\frac{1}{4}\sqrt{2T\log 2nT}\rceil$, gives the second part of the theorem. □

4. The FPL Meta-Algorithm for the Online Setting

In this section we overview the Follow the Perturbed Leader meta-algorithm for online learning and its application to the problem of online learning of eigenvectors. The meta-algorithm is given below. The algorithm relies on the availability of an oracle for the offline eigenvector problem, that is an oracle that given a matrix A, returns a leading eigenvector of A. We denote a call to this oracle by $\mathbf{EV}(A)$. Additionally, the algorithm relies on the availability of a distribution $\mathcal{D}$ over $\mathbb{S}_n$ from which it is possible to sample a perturbation matrix (efficiently). Different such distributions give rise to different instances of the algorithm with different regret guarantees.

Algorithm 3 Follow the Perturbed Leader

1: Input: distribution $\mathcal{D}$ over $\mathbb{S}_n$.
2: Sample a matrix $N\sim\mathcal{D}$.
3: $x_1\leftarrow\mathbf{EV}(N)$.
4: **for** $t=1,2,...$ **do**
5: Play x_t & observe A_t.
6: $x_{t+1}\leftarrow\mathbf{EV}\left(\sum_{\tau=1}^{t}A_\tau+N\right)$.
7: **end for**

Theorem 2. *The expected regret of algorithm 3 is upper bounded as follows.*

$$\mathbb{E}[\textit{regret}_T(\textit{FPL}(\mathcal{D}))]\leq\sum_{t=1}^{T}\mathbb{E}_{N\sim\mathcal{D}}[x_{t+1}^\top A_t x_{t+1}-x_t^\top A_t x_t]$$
$$+\mathbb{E}_{N\sim\mathcal{D}}[x_1^\top N x_1-x^{*\top}Nx^*]$$

where $x^*=\arg\max_{x\in\mathcal{S}}x^\top\left(\sum_{t=1}^{T}A_t\right)x$.

The proof is given in the appendix.

We now turn to survey two choices for the perturbation-generating distribution $\mathcal{D}$ and their corresponding regret bounds. We show that even though these distributions give rise to optimal algorithms in terms of the sequence length T, they suffer from a large dependence on the problem's dimension n.

4.1. Entry-wise Uniform Perturbation

Following (Kalai & Vempala, 2005) we consider the following entry-wise uniform distribution $\mathcal{D}_{uni}$. Each coordinate $i\geq j$ in the perturbation matrix N is sampled $\mathrm{U}[0,1/\epsilon]$ and for each coordinate $i<j$ we set $N_{i,j}\leftarrow N_{j,i}$ (in order for the resulting perturbation to be symmetric).

In (Kalai & Vempala, 2005) it was shown that with such noise distribution, one can bound the expected regret of a single round t as follows (see proof of Theorem 1.1. in (Kalai & Vempala, 2005)).

$$\mathbb{E}_{N\sim\mathcal{D}_{uni}}[x_{t+1}^\top A_t x_{t+1}-x_t^\top A_t x_t]\leq\epsilon\|A_t\|_1,$$

where $\|A\|_1=\sum_{i,j}|A_{i,j}|$.

Furthermore, since the sampled perturbation is bounded in ℓ_∞, using Holder's inequality we have that

$$\mathbb{E}_{N\sim\mathcal{D}_{uni}}[x_1^\top Nx_1-x^{*\top}Nx^*]$$
$$\leq\mathbb{E}_{N\sim\mathcal{D}_{uni}}[\|x_1x_1^\top-x^*x^{*\top}\|_1\cdot\|N\|_\infty]\leq\frac{D_1}{\epsilon},$$

where D_1 denotes the ℓ_1 diameter of the set $\{xx^\top\mid x\in\mathcal{S}\}$.

In order to derive the precise regret bound we need to bound both quantities $\|A_t\|_1, D_1$. The proof of the following lemma is given in the appendix.

Lemma 4. *It holds that $D_1=O(n)$ and for all t it holds that $\|A_t\|_1=O(n^{3/2})$. These bounds are also tight.*

Plugging the result of the lemma into Theorem 2 and optimizing over ϵ we have that

$$\mathbb{E}[\mathrm{regret}_T(\mathrm{FPL}(\mathcal{D}_{uni}))]=O\left(n^{5/4}\sqrt{T}\right). \tag{4}$$

4.2. Exponentially-distributed Perturbation

A second distribution that we consider, $\mathcal{D}_{exp}$, samples from $\mathbb{S}_n$ according to the following density.

$$d\mu(N)\propto\exp(-\epsilon\|N\|), \tag{5}$$

for appropriately chosen parameter ϵ.

A similar distribution was used for the problem of learning rotations (Hazan et al., 2010a) (altough in (Hazan et al., 2010a) the density was proportional to the nuclear norm and not the spectral as in Eq. (5)). For more details on how to sample from the distribution specified by Eq. (5) as well as a proof of the following lemma, the reader is referred to (Hazan et al., 2010b).

Lemma 5. *It holds that $\|N\|\sim$ Gamma$(n^2,1/\epsilon)$ and in particular $\mathbb{E}[\|N\|]=\frac{n^2}{\epsilon}$.*

The following lemma upper bounds the expected regret on a single round t.

Lemma 6. *On any time $t \in [T]$ it holds that*

$$\mathbb{E}_{N\sim\mathcal{D}_{exp}}[(x_{t+1}^\top A_t x_{t+1} - x_t^\top A_t x_t^\top] \leq e\epsilon.$$

The proof is given in the appendix.

Plugging Lemmas 5, 6 into Theorem 2 and optimizing over ϵ we have that

$$\mathbb{E}[\text{regret}_T(\text{FPL}(\mathcal{D}_{exp}))] = O\left(n\sqrt{T}\right). \quad (6)$$

5. New Perturbation and Analysis for FPL via Matrix Perturbation Theory

In this section we present our main result - a new noise distribution for the FPL algorithm and a corresponding analysis. In contrast to the analysis used in order to derive the regret bounds in Subsections 4.1, 4.2, which relies on geometric considerations and for which a large dependence of the regret bound on the problem's dimension n seems unavoidable, here we use a new analysis idea that is algebraic in nature and relies on tools from matrix perturbation theory which results in a much more moderate dependence on the dimension.

The new distribution, denoted $\mathcal{D}_{new}$, is based on a single parameter c and sampling from it is done as follows. We draw a vector $v \in \mathbb{R}^n$ whose entries are i.i.d. $\mathcal{N}(0,1)$ random variables and set the perturbation matrix to $N = c \cdot vv^\top$.

We prove the following theorem.

Theorem 3. *Let $c = \sqrt{\frac{T}{n}\max\{1, \ln(T/n)\}}$. Then*

$$\mathbb{E}[\textit{regret}_T(\textit{FPL}(\mathcal{D}_{new}))] = O(\sqrt{nT\max\{1, \ln(T/n)\}}).$$

Aside from the important improvement in the dependence on the dimension ($\sqrt{n}$ in Theorem 3 vs. at least n in Section 4), a key difference between the perturbations is in the efficiency of the resulting implementations. The key feature of the FPL algorithm for the online eigenvector problem is that the **EV**$(\cdot)$ oracle could be implemented using iterative methods such as the Power or Lanczos methods, that only require to compute matrix-vector products, to run in time that is typically $O(\text{nnz})$ where nnz denotes the sparsity of the input matrix. However, since the perturbations considered in Subsections 4.1. 4.2 are dense with high probability, even in case the support of all matrices A_t is sparse, the call to the oracle **EV** in Algorithm 3 will be with a dense matrix In contrast, the perturbation considered in this section is rank-one. Hence, while the perturbation matrix $N = cvv^\top$ is still dense with high probability, computing the product of N with a vector requires only $O(n)$ time which allows for an oracle implementation that could still benefit computationally from sparsity in the data.

We now turn to prove Theorem 3.

The following classic result in matrix perturbation theory is known as the Davis-Kahan *Sine Theroem*, see (Davis & Kahan, 1970). For ease of presentation, we restate the theorem and give a self-contained proof in the appendix.

Theorem 4 (Davis Kahan sine theorem)**.** *Let $A, B, E \in \mathbb{S}_n$ such that $B = A+E$ and assume that $\delta(A) > 0$. Denote by u_A the top eigenvector of A and by u_B the top eigenvector of B. It holds that*

$$\|u_A u_A^\top - u_B u_B^\top\|_F \leq 2\sqrt{2}\frac{\|E\|}{\delta(A)}.$$

The following theorem constitutes the key technical ingredient in the proof of Theorem 3. It upper bounds the cumulative distribution function of the eigengap of the perturbed matrix $A + cvv^\top$ for a given matrix A.

Theorem 5. *Let $A \in \mathbb{S}_n$ and let v be a vector of independent $\mathcal{N}(0,1)$ random variables and let c be a positive scalar. Denote $B = A + cvv^\top$. Then for any $\epsilon > 0$*

$$\Pr(\delta(B) \leq \epsilon) \leq \min\{\frac{2\sqrt{2}\epsilon}{\pi c}, 1\}.$$

The proof is based on anti-concentration results for the leading eigenvalue of the perturbed matrix. Due to its length and technical detail it is deferred to Appendix C.3. Here for some intuition, we prove a weaker version of Theorem 5, which captures some of the key ideas.

Lemma 7. *[Weaker version of Theorem 5] Let $A \in \mathbb{S}_n$ and let v be a vector of independent $\mathcal{N}(0,1)$ random variables and let c be a positive scalar. Denote $B = A + cvv^\top$. Then for any $\epsilon > 0$*

$$\Pr(\delta(B) \leq \epsilon) \leq \sqrt{\frac{2\epsilon}{\pi c}}.$$

Proof. Weyl's eigenvalues inequality states that for any two matrices $X, Y \in \mathbb{S}_n$ with eigenvalues $\lambda_1(X) \geq \lambda_2(X) \geq \ldots \geq \lambda_n(X)$ and $\lambda_1(Y) \geq \lambda_2(Y) \geq \ldots \geq \lambda_n(Y)$ it holds for any $i \in [n]$ that

$$\lambda_i(X) + \lambda_n(Y) \leq \lambda_i(X+Y) \leq \lambda_i(X) + \lambda_1(Y). \quad (7)$$

Applying Eq. (7) with $X = cvv^\top$, $Y = A$ and $i = 2$ gives us that

$$\lambda_2(B) \leq \lambda_2(cvv^\top) + \lambda_1(A) = \lambda_1(A),$$

where the equality follows since $vv^\top$ is rank-one.

Thus we have that

$$\delta(B) = \lambda_1(B) - \lambda_2(B) \geq \lambda_1(A + cvv^\top) - \lambda_1(A).$$

Let us denote by u_1 the eigenvector of A that corresponds to eigenvalue $\lambda_1(A)$. We continue to lower bound $\delta(B)$ as follows.

$$\delta(B) \geq u_1^\top \left(A + cvv^\top\right) u_1 - \lambda_1(A) = c(u_1^\top v)^2.$$

Since v is a vector of independent $\mathcal{N}(0,1)$ random variables we have that $(u_1^\top v)$ is also distributed as a $\mathcal{N}(0,1)$ random variable. Thus $(u_1^\top v)^2$ is a Chi-squared random variable with a single degree of freedom, also denoted as χ_1^2. If R is a χ_1^2 random variable then it holds that for any $\epsilon > 0$

$$\Pr(R \leq \epsilon) = \int_0^\epsilon \frac{e^{-t/2}}{\sqrt{2\pi t}} dt \leq \frac{1}{\sqrt{2\pi}} \int_0^\epsilon \frac{1}{\sqrt{t}} dt = \sqrt{\frac{2}{\pi}\epsilon}.$$

Thus we have that for any $\epsilon > 0$ it holds that

$$\Pr(\delta(B) \leq \epsilon) \leq \Pr\left((u_1^\top v)^2 \leq \frac{\epsilon}{c}\right) \leq \sqrt{\frac{2\epsilon}{\pi c}}.$$

□

The following lemma is a consequence of Theorem 5. The proof is given in the appendix.

Lemma 8. *Given $A \in \mathbb{S}_n$ let v be a random vector whose entries are independent $\mathcal{N}(0,1)$ random variables and let c be a positive scalar. Denote $\delta = \delta(A + cvv^\top)$. Define the random variable $X = \min(a, \delta^{-1})$ where a is a given positive constant. Then we have $\mathbb{E}[X] \leq \frac{2\sqrt{2}}{\pi c} \max\{\ln(\frac{\pi e a c}{2\sqrt{2}}), 1\}$.*

We are now ready to prove Theorem 3.

Proof. Starting from Theorem 2 we have that

$$\mathbb{E}[\text{regret}_T(\text{FPL}(\mathcal{D}_{new}))] \leq \sum_{t=1}^T \mathbb{E}_{N\sim\mathcal{D}_3}[x_{t+1}^\top A_t x_{t+1} - x_t^\top A_t x_t] + \mathbb{E}[x_1^\top N x_1 - x^{*\top} N x^*] \leq$$
$$\sum_{t=1}^T \mathbb{E}[\|x_{t+1}x_{t+1}^\top - x_t x_t^\top\|_* \|A_t\|] + \mathbb{E}[\|N\|] \leq$$
$$\sqrt{2}\sum_{t=1}^T \mathbb{E}[\|x_{t+1}x_{t+1}^\top - x_t x_t^\top\|_F] + c\mathbb{E}[\|v\|^2],$$

where the second inequality follows from Holder's inequality and the third follows from the fact that for any matrix in $\mathbb{S}_n$ with k non-zero eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k$ it holds that $\|A\|_F^2 = \sum_{i=1}^k \lambda_i^2 \geq \frac{1}{k}\left(\sum_{i=1}^k |\lambda_i|\right)^2 = \frac{1}{k}\|A\|_*^2$, and from our choice of the noise matrix N.

Denote $S_t = \sum_{\tau=1}^{t-1} A_\tau$. Since for any t, x_t is the leading eigenvector of the matrix $(S_t + N)$ and x_{t+1} is the leading eigenvector of the matrix $(S_{t+1} + N) = (S_t + N) + A_t$, we have by applying Theorem 4 that

$$\mathbb{E}[\|x_{t+1}x_{t+1}^\top - x_t x_t^\top\|_F] \leq \mathbb{E}[\min\{\sqrt{2}, 2\sqrt{2}\frac{\|A_t\|}{\delta(S_t+N)}\}]$$
$$\leq 2\sqrt{2}\mathbb{E}[\min\{\frac{1}{2}, \frac{1}{\delta(S_t+N)}\}],$$

where the min term is used since obviously $\|x_t x_t^\top - x_{t+1}x_{t+1}^\top\|$ is upper bounded by $\sqrt{2}$. The second inequality follows since $\|A_t\| \leq 1$.

Since all entries of v are $\mathcal{N}(0,1)$ random variables it holds that $\mathbb{E}[\|v\|^2] = n$ and thus

$$\mathbb{E}[\text{regret}_T(\text{FPL}(\mathcal{D}_{new}))] \leq$$
$$2\sqrt{2}\sum_{t=1}^T \mathbb{E}[\min\{\frac{1}{2}, \frac{1}{\delta(S_t + cvv^\top)}\}] + cn.$$

Applying the result of Lemma 8 with $a = \frac{1}{2}$ for every t gives

$$\mathbb{E}[\text{regret}_T(\text{FPL}(\mathcal{D}_{new}))] \leq$$
$$2\sqrt{2}T \cdot \frac{2\sqrt{2}}{\pi c}\max\{\ln(\frac{\pi e c}{4\sqrt{2}}), 1\} + cn.$$

Thus setting $c = \sqrt{\frac{T}{n}\max\{\ln(T/n), 1\}}$ yields the theorem. □

5.1. Using Approximate Eigenvector Computations

So far we have assumed that the eigenvector oracle used in Algorithm 3 finds an exact leading eigenvector. In practice, it is much more efficient to use iterative methods such as the Power and Lanczos algorithms to find an approximate eigenvector. The following theorem states that indeed Algorithm 3 admits such an efficient implementation, without sacrificing the regret guarantee given in Theorem 3. The proof is given in the appendix.

Theorem 6. *Algorithm 3, instantiated with noise distribution $\mathcal{D}_{new}$, admits an implementation such that the statement of Theorem 3 holds, and the worst-case time complexity of each iteration is $\tilde{O}(n^{-1/4}T^{3/4}nnz)$, where nnz denotes the joint-sparsity of all observed matrices $A_1, ..., A_T$.*

Acknowledgments

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 336078 – ERC-SUBLRN.

References

Arora, Sanjeev and Kale, Satyen. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, STOC*, 2007.

Balsubramani, Akshay, Dasgupta, Sanjoy, and Freund, Yoav. The fast convergence of incremental PCA. In *27th Annual Conference on Neural Information Processing Systems, NIPS*, 2013.

Cesa-Bianchi, Nicolò and Lugosi, Gábor. *Prediction, learning, and games*. Cambridge University Press, 2006.

Clarkson, Kenneth L., Hazan, Elad, and Woodruff, David P. Sublinear optimization for machine learning. *Journal of the ACM*, 59(5):23, 2012.

Davis, C. and Kahan, W. M. The rotation of eigenvectors by a perturbation, III. *SIAM J. Numer. Anal.*, 7, March 1970.

Garber, Dan and Hazan, Elad. Approximating semidefinite programs in sublinear time. In *25th Annual Conference on Neural Information Processing Systems 201*, pp. 1080–1088, 2011.

Grigoriadis, Michael D. and Khachiyan, Leonid G. A sublinear-time randomized approximation algorithm for matrix games. *Oper. Res. Lett.*, 18(2):53–58, 1995.

Hazan, Elad and Kale, Satyen. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML*, 2012.

Hazan, Elad, Kale, Satyen, and Warmuth, Manfred K. Learning rotations with little regret. In *23rd Conference on Learning Theory, COLT*, 2010a.

Hazan, Elad, Kale, Satyen, and Warmuth, Manfred K. Corrigendum to learning rotations with little regret. 2010b. URL http://ie.technion.ac.il/~ehazan/papers/rotfix.pdf.

Kalai, Adam Tauman and Vempala, Santosh. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

Kuczyński, J. and Woźniakowski, H. Estimating the largest eigenvalues by the power and lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13:1094–1122, October 1992.

Nie, Jiazhong, Kotlowski, Wojciech, and Warmuth, Manfred K. Online PCA with optimal regrets. In *24th International Conference on Algorithmic Learning Theory, ALT*, 2013.

Shamir, Ohad. A stochastic PCA algorithm with an exponential convergence rate. *CoRR*, abs/1409.2848, 2014. URL http://arxiv.org/abs/1409.2848.

Sylvester, J.J. On the relation between the minor determinants of linearly equivalent quadratic functions. *Philosophical Magazine Series 4*, 1(4):295–305, 1851.

Tropp, Joel A. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Tsuda, Koji, Rätsch, Gunnar, and Warmuth, Manfred K. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

Warmuth, Manfred K. and Kuzmin, Dima. Online variance minimization. In *19th Annual Conference on Learning Theory, COLT*, 2006a.

Warmuth, Manfred K. and Kuzmin, Dima. Randomized PCA algorithms with regret bounds that are logarithmic in the dimension. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, NIPS*, 2006b.

Zinkevich, Martin. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning, ICML*, 2003.