Stochastic PCA with ℓ_2 and ℓ_1 Regularization

Poorya Mianjy [1] **Raman Arora** [1]

Abstract

We revisit convex relaxation based methods for stochastic optimization of principal component analysis (PCA). While methods that directly solve the nonconvex problem have been shown to be optimal in terms of statistical and computational efficiency, the methods based on convex relaxation have been shown to enjoy comparable, or even superior, empirical performance – this motivates the need for a deeper formal understanding of the latter. Therefore, in this paper, we study variants of stochastic gradient descent for a convex relaxation of PCA with (a) ℓ_2, (b) ℓ_1, and (c) elastic net $(\ell_1 + \ell_2)$ regularization in the hope that these variants yield (a) better iteration complexity, (b) better control on the rank of the intermediate iterates, and (c) both, respectively. We show, theoretically and empirically, that compared to previous work on convex relaxation based methods, the proposed variants yield faster convergence and improve overall runtime to achieve a certain user-specified ϵ-suboptimality on the PCA objective. Furthermore, the proposed methods are shown to converge both in terms of the PCA objective as well as the distance between subspaces. However, there still remains a gap in computational requirements for the proposed methods when compared with existing nonconvex approaches.

1. Introduction

Principal component analysis (PCA), a ubiquitous procedure in scientific analysis, can be posed as the following learning problem: given a zero-mean random vector $\mathrm{x} \in \mathbb{R}^d$ with some (unknown) distribution $\mathcal{D}$, find the k-dimensional subspace that captures the maximal mass of the distribution. If we represent a subspace by an orthogonal basis matrix $\mathrm{U} \in \mathbb{R}^{d\times k}$ that spans the subspace, then PCA returns the subspace that maximizes the variance in data projected onto the subspace, i.e. $\mathbb{E}[\|\mathrm{U}^\top \mathrm{x}\|^2]$, among all k-dimensional subspaces of $\mathbb{R}^d$. Formally, we can write PCA as the following stochastic optimization problem:

$$\begin{array}{ll} \min\limits_{\mathrm{U}\in\mathbb{R}^{d\times k}} & -\mathbb{E}[\|\mathrm{U}^\top \mathrm{x}\|^2] \\ \text{subject to} & \mathrm{U}^\top\mathrm{U} = \mathrm{I}_k \end{array}. \tag{1}$$

Equivalently, we can represent a subspace by an orthogonal rank-k projection matrix $\mathrm{M} = \mathrm{U}\mathrm{U}^\top$, where U is any orthogonal basis matrix for the subspace. This gives the following equivalent formulation for the PCA problem:

$$\begin{array}{ll} \min\limits_{\mathrm{M}\in\mathbb{R}^{d\times d}} & -\mathbb{E}[\mathrm{x}^\top\mathrm{M}\mathrm{x}] \\ \text{subject to} & \mathrm{rank}(\mathrm{M}) = k,\ \lambda_i(\mathrm{M}) \in \{0,1\}, \forall i \in [d] \end{array} \tag{2}$$

It is easy to check that this maximal variance subspace is given by the span of top-k eigenvectors of the covariance matrix $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$. In other words given eigendecomposition of $\mathrm{C} = \sum_{i=1}^d \lambda_i \mathrm{u}_i\mathrm{u}_i^\top$, for $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, the optimal solution to Problem 1 is given by the basis matrix $\mathrm{U}_* = [\mathrm{u}_1, \ldots, \mathrm{u}_k]$, and the optimal solution to Problem 2 is given as $\mathrm{M}_* = \mathrm{U}_*\mathrm{U}_*^\top$. Furthermore, given access to the distribution $\mathcal{D}$ only through a sample $\{\mathrm{x}_i\}_{i=1}^n \sim \mathcal{D}^n$ drawn independently from $\mathcal{D}$, the sample average approximation (SAA, or equivalently empirical risk minimization) approach to learning the maximal variance subspace amounts to finding the top-k eigenvectors of the empirical covariance matrix, $\widehat{\mathrm{C}} := \frac{1}{n}\sum_{i=1}^n \mathrm{x}_i\mathrm{x}_i^\top$.

An alternative computationally attractive approach to solving Problem 1 is based on stochastic approximation (SA) algorithms that attempt to directly minimize the objective given access to a first order oracle. For instance, stochastic gradient descent (SGD), a staple SA algorithm, on Problem 1 yields the following updates:

$$\mathrm{U}_{t+1} = \texttt{Gram-Schmidt}(\mathrm{U}_t + \eta_t \mathrm{x}_t\mathrm{x}_t^\top \mathrm{U}_t), \tag{3}$$

where Gram-Schmidt orthogonalization gives an orthogonal basis matrix for the column span of $\mathrm{U}_t + \eta_t\mathrm{x}_t\mathrm{x}_t^\top\mathrm{U}_t$; this is also known as Oja's algorithm (Oja, 1982). Such first order methods for solving Problem 1 have received a lot of attention in recent years (Arora et al., 2012; 2013; Balsubramani et al., 2013; Mitliagkas et al., 2013; Shamir,

[1] Department of Computer Science, Johns Hopkins University, Baltimore, USA. Correspondence to: Raman Arora <arora@cs.jhu.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

2016; Allen-Zhu & Li, 2017; Jain et al., 2016; Balcan et al., 2016). It is remarkable that even though Problem 1 is nonconvex[1], Oja's algorithm works reasonably well in practice and has been shown to enjoy strong theoretical guarantees (Allen-Zhu & Li, 2017; Jain et al., 2016; Balcan et al., 2016; Shamir, 2016).

Rather than directly solve the nonconvex problem, one can consider a convex relaxation of the equivalent formulation in Problem 2. Following Arora et al. (2013), we take the convex hull of the constraint set in Problem 2 to obtain the following convex program:

$$\begin{aligned} \min_{\mathrm{M}\in\mathbb{R}^{d\times d}} \quad & -\mathbb{E}[\mathrm{x}^\top \mathrm{M}\mathrm{x}] \\ \text{subject to} \quad & \mathrm{Tr}\left(\mathrm{M}\right) = k,\ 0 \preceq \mathrm{M} \preceq \mathrm{I} \end{aligned} . \tag{4}$$

Stochastic gradient descent on Problem 4 yields the following update, also referred to as matrix stochastic gradient (MSG) in (Arora et al., 2013):

$$\mathrm{M}_{t+1} \leftarrow \mathscr{P}(\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top), \tag{5}$$

where $\mathscr{P}(\cdot)$ is the projection operator onto the feasible set of Problem 4 with respect to the Frobenius norm. Assuming the fourth moment of the distribution is bounded by a constant, the standard analysis of (Shamir & Zhang, 2013) yields the following guarantee for MSG:

$$\mathbb{E}[\mathrm{x}^\top \mathrm{M}_* \mathrm{x}] - \mathbb{E}[\mathrm{x}^\top \bar{\mathrm{M}} \mathrm{x}] = O\left(\frac{\log T}{\sqrt{T}}\right),$$

where M_* is the optimal solution to the PCA Problem 2 and $\bar{\mathrm{M}} = \texttt{rounding}(\mathrm{M}_{T+1})$ is a rank-k projection matrix obtained using randomized rounding (Warmuth & Kuzmin, 2008) of the final iterate of MSG.

Compared to Oja's algorithm, MSG has two major drawbacks: first, Oja's algorithm achieves a faster $\tilde{O}(\frac{1}{T})$ rate of convergence, and second, Oja's algorithm is computationally efficient since at each iteration it only keeps a $d \times k$ orthogonal matrix, while rank of the projection matrix M_t in MSG can possibly grow at each iteration.

More generally, methods based on *convex relaxation* are usually hard to scale to large problems due to both statistical inefficiency and higher computational cost. On the other hand, methods that directly solve the non-convex problems are usually preferable by practitioners, and have been shown recently to achieve optimal convergence rates in many problems. In particular, the classical Oja's algorithm which is simply SGD on the original non-convex PCA problem is provably optimal both statistically and computationally. However, empirically, it has been shown that convex relaxation based methods either match or outperform Oja's performance (see for example (Arora et al., 2012; 2013)). Thus, a natural question to ask is if the suboptimal guarantees on statistical and computational efficiency of methods based on convex relaxation are artifacts of analysis.

To understand these issues better, in this paper we study various modifications to the MSG update in equation (4) that impart the resulting algorithm with desirable computational properties including faster convergence and better overall runtime. These modifications are based on principled design techniques – each of the proposed variants is given as stochastic gradient descent on the the MSG objective in Problem 4 with an additional regularization term. In particular, we consider (a) an ℓ_2 regularization which yields a faster convergence rate, (b) an ℓ_1 regularization term which prevents the rank of the intermediate iterates of MSG from growing unbounded, and (c) an elastic-net (i.e. joint $\ell_1 + \ell_2$) regularization that is empirically shown to achieve a faster convergence rate with small computational cost per iteration. Our experimental results show that the proposed $(\ell_1 + \ell_2)$-regularized MSG achieves state-of-the-art results outperforming MSG and Oja's algorithm for various parameter settings and choice of datasets.

It is important to note that we are interested in principled design and analysis of stochastic approximation algorithms for principal component analysis (i.e. Problems 1 and 2). The proposed formulations of the regularized PCA objective are purely for computational reasons unlike usual regularized learning problems where the goal is to avoid overfitting (equivalently, injecting an inductive bias).

1.1. Notation

We denote matrices with capital Roman letters, e.g. U, and vectors with small Roman letters, e.g. u. For any integer k, we denote the set $\{1, \dots, k\}$ by $[k]$. Furthermore, I_k denotes the identity matrix of size $k \times k$. We drop the subscript k whenever the size is clear from the context. Frobenius norm and spectral norm of matrices are denoted by $\|\cdot\|_F$ and $\|\cdot\|_2$ resepctively, and for vectors, $\|\cdot\|$ denotes the ℓ_2 norm. For any two matrices $\mathrm{M}_1, \mathrm{M}_2 \in \mathbb{R}^{d\times d}$, the standard inner-product is written as $\langle \mathrm{M}_1, \mathrm{M}_2 \rangle = \mathrm{Tr}\left(\mathrm{M}_1^\top \mathrm{M}_2\right)$. For a real symmetric matrix C, $\lambda_k(\mathrm{C})$ represents the k^{th} largest eigenvalue of C. For notational convenience, when clear from the context we will denote $\lambda_k(\mathrm{C})$ by λ_k. We denote the eigen-decomposition of the covariance matrix by $\mathrm{C} = \sum_{i=1}^d \lambda_i \mathrm{u}_i \mathrm{u}_i^\top$, where $\lambda_1 \geq \cdots \geq \lambda_d$. For any $i \in [d-1]$, we denote the eigengap at i as $g_i := \lambda_i - \lambda_{i+1}$. The projection matrix onto the subspace spanned by the top-k eigenvectors of C will be represented as $\Pi_k(\mathrm{C}) = \sum_{i=1}^k \mathrm{u}_i \mathrm{u}_i^\top$. The convex hull of the set of rank-k orthogonal projection matrices is denoted by $\mathscr{M} := \{\mathrm{M} \in \mathbb{R}^{d\times d} : \mathrm{Tr}\left(\mathrm{M}\right) \leq k, 0 \preceq \mathrm{M} \preceq \mathrm{I}\}$. Finally, our analysis for regularized variants leverages the strong

[1] since the objective is concave and the set of orthogonal matrices is non-convex

Algorithm 1 ℓ_2-Regularized MSG (ℓ_2-RMSG)

```
Require: Input data $\{x_t\}_{t=1}^T$, output dimension $k$, regularization parameter $\lambda$
Ensure: $\tilde{M}$
1: $M_1 \leftarrow 0$
2: for $t = 1, \cdots, T$ do
3:     $\eta_t \leftarrow \frac{1}{\lambda t}$
4:     $M_{t+\frac{1}{2}} \leftarrow (1 - \lambda\eta_t)M_t + \eta_t x_t x_t^\top$
5:     $M_{t+1} \leftarrow \mathscr{P}_{\mathscr{M}}\left(M_{t+\frac{1}{2}}\right)$
6: end for
7: $\tilde{M} \leftarrow \mathscr{P}_{\text{rank}-k}(M_{T+1})$ {Return top-$k$ subspace of $M_{T+1}$}
```

convexity / smoothness of the corresponding objectives.

Definition 1.1 (Strongly convex / smooth)**.** A function $f : \mathscr{M} \to \mathbb{R}$ is λ-strongly convex and μ-smooth for some $\lambda, \mu > 0$ if for all $M, M' \in \mathscr{M}$ and $g_M := \nabla f(M)$, we have

$$\frac{\lambda}{2}\|M'-M\|_F^2 \leq f(M') - f(M) - \langle g_M, M'-M\rangle \leq \frac{\mu}{2}\|M'-M\|_F^2.$$

2. PCA with ℓ_2-regularization

In this section, we study how adding a strongly convex regularizer to the linear PCA objective in Problem 4 changes the optimization problem and if we can leverage strong convexity of the resulting objective to guarantee a faster convergence rate without changing the optimum. We consider the following ℓ_2-regularized PCA optimization problem:

$$\begin{aligned} \min_{M \in \mathbb{R}^{d \times d}} \quad & -\mathbb{E}[x^\top M x] + \frac{\lambda}{2}\|M\|_F^2 \\ \text{subject to} \quad & \mathrm{Tr}\,(M) \leq k,\ 0 \preceq M \preceq I \end{aligned}. \tag{6}$$

SGD on Problem 6 yields the following ℓ_2-RMSG updates,

$$M_{t+1} \leftarrow \mathscr{P}\left((1-\lambda\eta_t)M_t + \eta_t x_t x_t^\top\right). \tag{7}$$

Note that as is the case with MSG, the final iterate, M_{T+1}, of ℓ_2-RMSG is not guaranteed to be a rank-k projection matrix. As we show below, we can guarantee convergence in terms of the distance between the subspaces, which allows us to perform a simple deterministic rounding by simply returning the top-k eigenspace of M_{T+1} (see proof of Theorem 2.4). This yields the procedure detailed in Algorithm 1.

Next, we study the iteration complexity of ℓ_2-RMSG and conditions on the size of the regularization constant that keep the optimum unchanged. Before giving formal results, we make a few remarks summarizing the key observations.

Fast rate: The objective in Problem 6 is λ-strongly convex. We leverage this to show that ℓ_2-RMSG enjoys a fast rate of $O\left(\frac{1}{\lambda^2 T}\right)$; see Theorem 2.4 for a formal statement.

Admissible λ: While the bound above suggests that larger values of λ are preferred, it is important to note that for large λ the optima of the Problems 2 and 6 may fail to coincide. Our analysis shows that if there exists an eigengap at k in the spectrum of the covariance matrix, i.e. if $g_k = \lambda_k(C) - \lambda_{k+1}(C) > 0$, then for $\lambda < g_k$, the global optimum of the regularized PCA problem is a global optimum for the original problem, even when there is no eigengap at k – we refer to such values of the regularization parameter as *admissible*; see Section 2.1 for more details. For illustration, we plot the landscape of the ℓ_2-regularized objective for different values of λ in Figure 1(a) and (b) for covariance matrices with and without an eigengap at k, respectively.

Overall runtime: The ℓ_2-RMSG algorithm has the same computational cost per iterate as MSG. To see this, note that the Step 4 of Algorithm 1 can be written equivalently in terms of eigen-decomposition of $M_t = U_t\Lambda_t U_t^\top$ as follows,

$$\begin{aligned} &(1-\lambda\eta_t)U_t\Lambda_t U_t^\top + \eta_t x_t x_t^\top \\ &\quad = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \underbrace{\begin{bmatrix} (1-\lambda\eta_t)\Lambda_t + \hat{x}_t\hat{x}_t^\top & \|r_t\|\,\hat{x}_t \\ \|r_t\|\,\hat{x}_t^\top & \|r_t\|^2 \end{bmatrix}}_{Q_t \in \mathbb{R}^{(l+1)\times(l+1)}} \begin{bmatrix} U_t^\top \\ \frac{r_t^\top}{\|r_t\|} \end{bmatrix} \end{aligned}$$

where $\hat{x}_t = \sqrt{\eta_t}U_t^\top x_t$ and $r_t = \sqrt{\eta_t}x_t - U_t\hat{x}_t$. Therefore, the update in Step 4 amounts to computing the eigendecompostion of matrix $Q_t = \tilde{U}\tilde{\Lambda}\tilde{U}$ and matrix multiplication, $U_{t+1} = [U_t \ \ \frac{r_t}{\|r_t\|}]\tilde{U}$. This rank-one eigenupdate requires $O(k_t^3 + dk_t^2)$ time and $O(dk_t)$ space where $k_t = \text{rank}(M_t)$; this computational trick is well-known in literature (Arora et al., 2013). The projection in Step 5 of Algorithm 1 can be implemented efficiently as well via a simple shift-and-cap procedure, Algorithm 2 of (Arora et al., 2013), on the vector consisting of diagonal entries of $\tilde{\Lambda}$; this requires $O(k_t \log k_t)$ time.

2.1. Admissible values of regularization parameter

We begin with a formal definition of admissibility.

Definition 2.1. We say that the regularization parameter λ takes an admissible value, if any optimum of the regularized PCA Problem 6 is also an optimum of the original Problem 4.

In what follows, we give sufficient conditions on admissible values of λ under two distinct settings – with and without an eigengap at k. We define the eigengap at k as $g_k := \lambda_k(C) - \lambda_{k+1}(C)$. We say that there exists an eigengap at k, if $g_k > 0$, otherwise (i.e. if $g_k = 0$), we say that there is no eigengap at k. When we have an eigengap at k, we can give the following simple characterization of admissible values for the regularization parameter λ. All proofs are deferred to the Appendix in the supplementary file.

Lemma 2.2 (Admissible λ with an eigengap at k)**.** Let $g_k > 0$. Then, for any regularization parameter $0 \leq \lambda < g_k$,

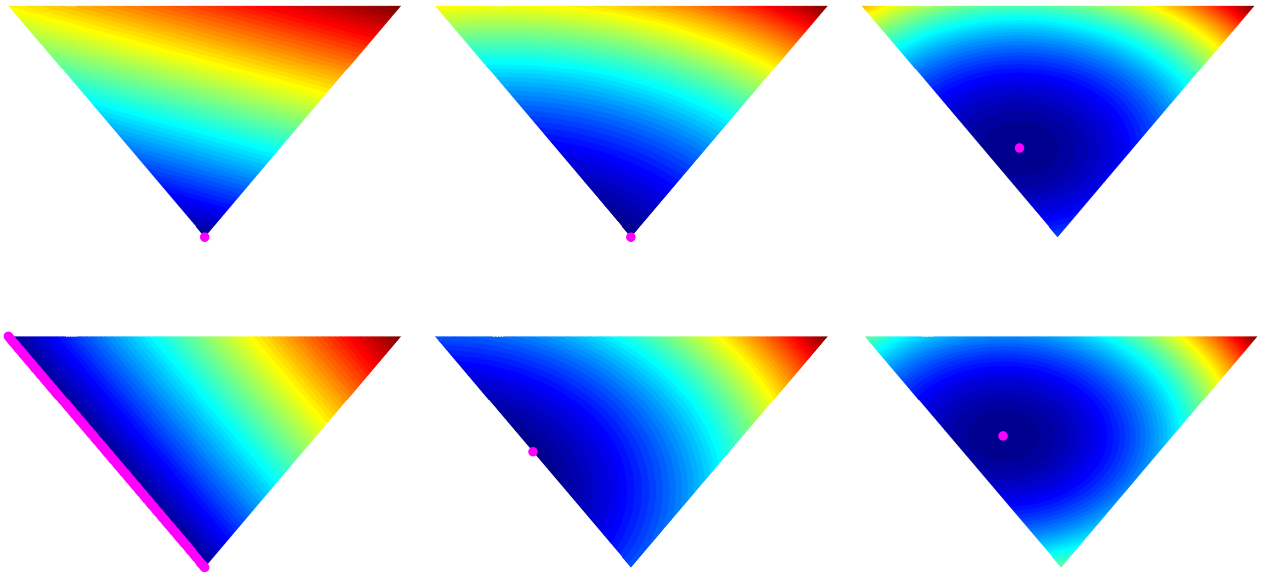

Figure 1: Geometry of the optimization problem (6) with (top) and without (bottom) an eigengap at k. The feasible region of Problem 4, is the simplex shown above (for $d = 3$ and $k = 2$). Each vertex represents a rank-2 projection matrix. The optima is marked with magenta circles. **Left**: the linear objective of the original PCA problem. The optimum is attained at the lower vertex (top) and the left edge (bottom) of the feasible triangle. **Middle**: objective of the regularized PCA problem, with a sufficiently small amount of curvature (admissible λ). The optimum is still an optimum of the original problem. **Right**: objective of the regularized PCA problem with too big of a curvature. The regularizer has deformed the objective and altered the optimum of the original problem.

the optimum of the original PCA Problem (6) and the ℓ_2-regularized PCA Problem (4) are unique and identical.

Next, we consider the case when there is no eigengap at k, i.e. $g_k = 0$. We do assume that an eigengap exists somewhere in the spectrum, since otherwise the covariance matrix is simply a multiple of the identity matrix which is not an interesting case – any k-dimensional subspace is an optimum of the corresponding PCA problem. Without loss of generality, assume that rank(C) $> k$. Moreover, lets denote $\lambda_0(\mathrm{C}) := +\infty$ and $\lambda_{d+1}(\mathrm{C}) := 0$ for notational convenience. Then, we have the following result.

Lemma 2.3 (Admissible λ without an eigengap at k)**.** Let p and q be respectively the largest index smaller than k and the smallest index larger than k, at which C has an eigengap:

$$p := \max\{i : i \in \{0, \dots, k-1\},\ \lambda_i > \lambda_{i+1}\}$$
$$q := \min\{i : i \in \{k+1, \dots, d\},\ \lambda_i > \lambda_{i+1}\},$$

Then, for $0 < \lambda < \min\{g_p, g_q\}$, the optimum of Problem 6 is uniquely given by

$$\mathrm{M}_* = \sum_{i=1}^{p} \mathrm{u}_i \mathrm{u}_i^\top + \frac{k-p}{q-p} \sum_{j=p+1}^{q} \mathrm{u}_j \mathrm{u}_j^\top .$$

Furthermore, M_* is an optimum solution to Problem 4.

In order to better understand the expression for the global optimum, M_*, in Lemma 2.3, consider the following. If there is no eigengap at k, then there is a *maximal* subset of indices $\mathscr{S} := \{p+1, \dots, q\} \subseteq [d]$ such that $k \in \mathscr{S}$ and $\lambda_{p+1} = \cdots = \lambda_q$. In terms of the variance captured, there is no advantage in choosing any particular (convex combination) of the rank-1 subspaces associated with the eigenvectors indexed by $\mathscr{S}$. However, to minimize the Frobenius norm penalty, ℓ_2-RMSG picks the average of these subspaces. This is vivid in the expression for M_* – the top-p rank-1 subspaces are included in M_*, and the remaining $k-p$ mass is distributed equally among the $q-p$ rank-1 subspaces indexed by $\mathscr{S}$. Figure 1 provides a geometric illustration. As can be seen in the bottom row, all points on the left edge are equally good in terms of the objective. But, the ℓ_2-regularizer chooses the average of the subspaces with equal eigenvalues, which minimizes the ℓ_2-penalty.

2.2. Convergence Analysis of ℓ_2-RMSG

Our main result of this section bounds the sub-optimality of the output of Algorithm 1, for admissible values of the regularization parameter λ, in terms of the distance from the optimal subspace, M_*. We show the convergence in terms of the parameter as well as a faster rate as compared with MSG (Arora et al., 2013).

Theorem 2.4. Assume $\mathbb{E}[\|\mathrm{x}\|^2] \leq 1$. Then, for any admissible value of λ, after T iterations of Algorithm 1 starting at

$\mathrm{M}_1 = 0$, with step-size sequence $\eta_t = \frac{1}{\lambda t}$, we have that

$$\mathbb{E}[\|\tilde{\mathrm{M}} - \mathrm{M}_*\|_F^2] \le \frac{16(1+\lambda\sqrt{k})^2}{\lambda^2 T}, \tag{8}$$

where M_* is an optimum of Problem 4, $\tilde{\mathrm{M}}$ is the output after rounding the final iterate to a rank-k matrix, and the expectation is with respect to the distribution $\mathcal{D}$.

Minimax Optimality: Theorem 2.4 guarantees that the iterates of Algorithm 1 converge to the optimal projection matrix in Frobenius norm. It is easy to see that this metric is directly related to the angle between subspaces, i.e. to $\|\tilde{\mathrm{U}}^\top \mathrm{U}_*\|_F^2$,

$$\|\mathrm{M}_* - \tilde{\mathrm{M}}\|_F^2 = 2\left(k - \|\tilde{\mathrm{U}}^\top \mathrm{U}_*\|_F^2\right),$$

where $\tilde{\mathrm{M}} = \tilde{\mathrm{U}}\tilde{\mathrm{U}}^\top$ and $\mathrm{M}_* = \mathrm{U}_*\mathrm{U}_*^\top$. This provides a basis for comparison against previous works of Shamir (2016) and Allen-Zhu & Li (2017) which measure convergence in terms of the angle between the subspaces. Furthermore, as a corollary of Theorem 2.4, we have the following bound in terms of the angle between subspaces for ℓ_2-RMSG,

$$\mathbb{E}[\|\tilde{\mathrm{U}}^\top \mathrm{U}_*\|_F^2] \le k - \frac{8(1+\lambda\sqrt{k})^2}{\lambda^2 T},$$

which is minimax optimal in an information-theoretic sense (see Theorem 6 of Allen-Zhu & Li (2017)).

Finally, convergence in parameter implies the following guarantee in terms of PCA objective for ℓ_2-RMSG.

Theorem 2.5. Under same assumptions as Theorem 2.4, we have that after T iterations of Algorithm 1,

$$\mathbb{E}[\mathrm{x}^\top \mathrm{M}_* \mathrm{x} - \mathrm{x}^\top \tilde{\mathrm{M}} \mathrm{x}] \le \frac{8\lambda_1(1+\lambda\sqrt{k})^2}{\lambda^2 T}. \tag{9}$$

A proof of Theorem 2.5 is provided in the supplementary.

3. PCA with ℓ_1-Regularization

As discussed above, the overall runtime needed for MSG to find an ϵ-suboptimal solution depends critically on the rank, k_t, of the intermediate iterates, M_t. If k_t is as large as d, then MSG achieves a runtime that is cubic in the dimensionality. In order to overcome this computational barrier, which appears to be a natural artifact of convex relaxations, we consider regularizing the PCA objective with an ℓ_1 penalty. In particular, we consider the following problem in this section:

$$\begin{array}{ll} \min\limits_{\mathrm{M}\in\mathbb{R}^{d\times d}} & -\mathbb{E}_{\mathrm{x}}[\mathrm{x}^\top \mathrm{M}\mathrm{x}] + \mu \mathrm{Tr}(\mathrm{M}) \\ \text{subject to} & \mathrm{Tr}(\mathrm{M}) \le k,\ 0 \preceq \mathrm{M} \preceq \mathrm{I} \end{array}. \tag{10}$$

Algorithm 2 ℓ_1-Regularized MSG (ℓ_1-RMSG)

```
Require: Input data $\{\mathrm{x}_t\}_{t=1}^T$, output dimension $k$, regularization parameter $\mu$
Ensure: $\tilde{\mathrm{M}}$
$\mathrm{M}_1 \leftarrow 0$
for $t = 1, \ldots, T$ do
    $\eta_t \leftarrow \frac{2}{1+\mu\sqrt{d}}\sqrt{\frac{k}{t}}$
    $\mathrm{M}_{t+\frac{1}{2}} \leftarrow \mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top - \mu\eta_t \mathrm{I}$
    $\mathrm{M}_{t+1} \leftarrow \mathcal{P}_{\mathcal{M}}\left(\mathrm{M}_{t+\frac{1}{2}}\right)$
end for
$\tilde{\mathrm{M}} \leftarrow$ rounding($\mathrm{M}_{T+1}$) {Algorithm 2 of (Warmuth & Kuzmin, 2008)}
```

The objective in Problem 10 is a linear function in M. Projected gradient descent on this problem yields the following updates:

$$\mathrm{M}_{t+1} = \mathcal{P}_{\mathcal{M}}(\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top - \eta_t \mu \mathrm{I}), \tag{11}$$

where $\mathcal{P}_{\mathcal{M}}(\cdot)$ projects onto the feasible set $\mathcal{M}$ with respect to the Frobenius norm. This gives the ℓ_1-RMSG procedure described in Algorithm 2. The design rationale motivating the ℓ_1 penalty is that it promotes low-rank iterates, thereby controlling computational cost per iteration. To see this, note that each update in equation (11) involves a shift by $-\mu\eta_t\mathrm{I}$, which shrinks the spectrum of $\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top$ by $\mu\eta_t$. In other words, the value $\mu\eta_t$ will serves as a cut-off parameter that will zero out any eigenvalue smaller than $\mu\eta_t$.

Admissible μ. As in the previous section, we are interested in μ such that the regularized problem has the same optimum as the original problem. Formally, we say that the regularization parameter μ takes an admissible value, if any solution to the regularized PCA Problem 10 is also a solution to the original Problem 4. The following lemma gives a sufficient condition on the admissibility of μ.

Lemma 3.1. Let $g_k > 0$. Then, for any regularization parameter $0 \le \mu \le \lambda_k(\mathrm{C})$, the optimum of Problem 10 is unique, and is an optimum for Problem 4.

Key insights. KKT first-order optimality condition on Problem 10 gives $\lambda_k(\mathrm{C}) = \mu - \gamma_k + \omega_k + \beta$, where $\gamma_k, \omega_k, \beta \ge 0$ are Lagrange multipliers associated with the constraints $\lambda_k(\mathrm{M}) \ge 0$, $\lambda_k(\mathrm{M}) \le 1$ and $\mathrm{Tr}\,\mathrm{M} \le k$. If $\mu > \lambda_k(\mathrm{C})$, and since $\beta, \omega_k \ge 0$, it should hold that $\gamma_k > 0$. By complementary slackness (i.e. $\gamma_k \lambda_k(\mathrm{M}^*) = 0$), we conclude that $\lambda_k(\mathrm{M}^*) = 0$. In this case, M^* cannot be a solution to Problem 4. This further implies that the condition in Lemma 3.1 is also necessary.

3.1. Convergence Analysis of ℓ_1-RMSG

Our first main result of this section gives a bound on the suboptimality of Algorithm 2 in terms of the PCA objective.

Theorem 3.2. Assume $\mathbb{E}[\|\mathrm{x}\|^2] \le 1$. Then, for any admissible regularization parameter $\mu \le \min\{\frac{\lambda_k}{2}, \frac{1}{\sqrt{d}}\}$, after T iterations of Algorithm 2 with step size $\eta_t = \frac{2}{1+\mu\sqrt{d}}\sqrt{\frac{k}{t}}$, and starting at $\mathrm{M}_1 = 0$, we have that:

$$\mathbb{E}[\mathrm{x}^\top \mathrm{M}_* \mathrm{x}] - \mathbb{E}[\mathrm{x}^\top \tilde{\mathrm{M}} \mathrm{x}] \le \frac{64\sqrt{k}\log T}{\sqrt{T}},$$

where M_* is an optimum of Problem 4, $\tilde{\mathrm{M}}$ is the output after `rounding`, and the expectation is with respect to the distribution $\mathscr{D}$ and the randomization in the algorithm.

The result above shows that for sufficiently small admissible μ, ℓ_1-RMSG converges at the same rate as MSG, i.e. $O(\frac{\log T}{\sqrt{T}})$. However, we argue in the next section that in terms of the overall runtime, ℓ_1-RMSG has an advantage. We show formally that the ℓ_1 regularization prevents the rank of ℓ_1-RMSG iterates from growing too large.

3.2. Rank Control for ℓ_1-RMSG

To get a handle on the rank of the iterates, we first need to show that iterates have a small tail, i.e. the bottom eigenvalues of M_t are small enough to get eliminated by the shrinkage step. The following lemma formalizes this intuition.

Lemma 3.3. For all $t = 1, \dots, T$, it holds that

$$\sum_{i=k+1}^{d} \lambda_i(\mathrm{M}_t) \le \frac{1}{g_k}\langle \mathrm{C} - \mu \mathrm{I}, \mathrm{M}_* - \mathrm{M}_t \rangle,$$

where $g_k = \lambda_k - \lambda_{k+1}$ is the eigengap at k.

Next, we need to show that if the tail of M_t is small, then the update $\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top$ will also have a small tail.

Lemma 3.4. For any iterate $t \in \{1, \dots, T\}$, it holds that

$$\mathbb{E}[\sum_{i=k+1}^{d} \lambda_i(\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top)] \le \sum_{i=k+1}^{d} \lambda_i(\mathrm{M}_t) + \eta_t.$$

Finally, we argue that shrinking the spectrum by $-\mu\eta_t \mathrm{I}$ will likely eliminate bottom eigenvalues of the update $\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top$, resulting in a low rank iterate. Formally, our second main result of this section states the following.

Theorem 3.5. Under the same assumptions as Theorem 3.2, for any admissible μ, we have that for all iterates

$$\mathbb{E}[\mathrm{rank}(\mathrm{M}_{t+1})] \le k + \frac{33 \log t}{\mu g_k}$$

Note that since $\mu \le 1/\sqrt{d}$, it is easy to check that $\mathbb{E}[\mathrm{rank}(\mathrm{M}_{t+1})] \le k + \tilde{O}(\sqrt{d}/g_k)$. Theorem 3.5 together with Theorem 3.2 demonstrates an interesting regime, where MSG and ℓ_1-RMSG show exact same statistical convergence behavior, but ℓ_1-RMSG iterates are guaranteed to have a significantly smaller rank than the input dimension.

Algorithm 3 $\ell_2 + \ell_1$-Regularized MSG ($\ell_{2,1}$-RMSG)

```
Require: $\{\mathrm{x}_t\}_{t=1}^T,\ k, \lambda, \mu$
Ensure: $\tilde{\mathrm{M}}$
$\mathrm{M}_1 \leftarrow 0$
for $t = 1, \dots, T$ do
    $\eta_t \leftarrow \frac{1}{\lambda t}$
    $\mathrm{M}_{t+\frac{1}{2}} \leftarrow (1 - \lambda\eta_t)\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top - \mu\eta_t \mathrm{I}$
    $\mathrm{M}_{t+1} \leftarrow \mathscr{P}_{\mathscr{M}}\left(\mathrm{M}_{t+\frac{1}{2}}\right)$
end for
$\tilde{\mathrm{M}} \leftarrow \mathscr{P}_{\text{rank}-k}(\mathrm{M}_{T+1})$ {Return top-$k$ subspace of $\mathrm{M}_{T+1}$}
```

4. PCA with Elastic-net Regularization

We saw in the previous sections that ℓ_2-RMSG for solving PCA with Frobenius norm regularization enjoys a faster convergence rate whereas ℓ_1-RMSG for solving PCA with trace norm regularization is better in terms of computational cost per iteration. In this section, we propose a variant that combines both the ℓ_1 and ℓ_2 regularization in the hope that it yields the best of both worlds, i.e. simultaneously provide fast rate with rank-control. We consider PCA with the following *elastic-net* regularization:

$$\begin{array}{ll} \min\limits_{\mathrm{M}\in\mathbb{R}^{d\times d}} & -\mathbb{E}_{\mathrm{x}}[\mathrm{x}^\top \mathrm{M} \mathrm{x}] + \mu \mathrm{Tr}(\mathrm{M}) + \frac{\lambda}{2}\|\mathrm{M}\|_F^2 \\ \text{subject to} & \mathrm{Tr}(\mathrm{M}) \le k,\ 0 \preceq \mathrm{M} \preceq \mathrm{I} \end{array}. \tag{12}$$

SGD on Problem 12 yields the following updates:

$$\mathrm{M}_{t+1} = \mathscr{P}_{\mathscr{M}}((1 - \lambda\eta_t)\mathrm{M}_t + \eta_t \mathrm{x}_t \mathrm{x}_t^\top - \mu\eta_t \mathrm{I}), \tag{13}$$

where $\mathscr{P}_{\mathscr{M}}(\cdot)$ projects onto the feasible set $\mathscr{M}$ w.r.t the Frobenius norm; detailed procedure is given in Algorithm 3.

Again, we should be judicious in our choice of the regularization parameters μ and λ so as to ensure that the regularized problem has the same optimum as the original problem; the following result provides a sufficient condition.

Lemma 4.1 (Admissibility of (λ, μ))**.** Assume that $g_k > 0$. Then, for any pair of regularization parameters λ and μ such that $0 < \lambda < g_k$ and $0 \le \lambda + \mu \le \lambda_k$, the optimum of Problem 12 is unique, and is an optimum for Problem 4. We call any such (λ, μ)-pair an admissible regularization pair.

We conclude this section by noting that it is straightforward to adapt Theorem 2.4 to get a faster $O(\frac{1}{\epsilon})$ iteration complexity for $\ell_{2,1}$-RMSG, as long as we choose learning rate $\eta_t = O(\frac{1}{t})$. Also, one can show rank-control as in Theorem 3.5, under the learning rate $\eta_t = O(\frac{1}{\sqrt{t}})$. It would be desirable to guarantee both faster statistical rates and computational cost per iteration simultaneously. Our current analysis falls short in establishing such a result. However, in Section 5, we provide empirical evidence to support statistical and computational benefits of $\ell_{2,1}$-RMSG.

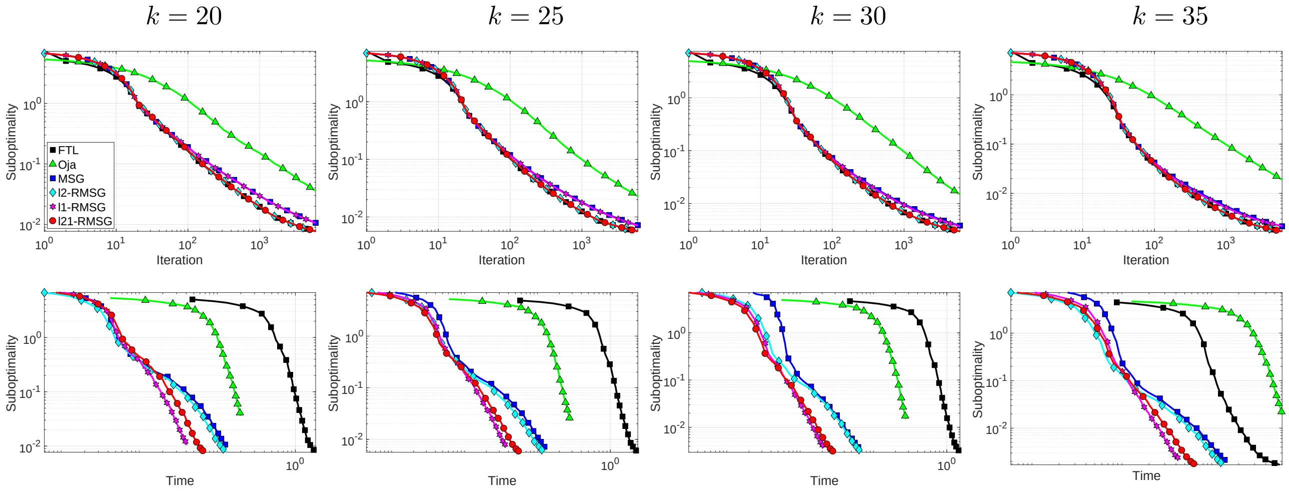

Figure 2: Comparisons of FTL, MSG, ℓ_2-RMSG, ℓ_1-RMSG, $\ell_{2,1}$-RMSG and Oja's algorithm on a synthetic dataset, in terms of the suboptimality as a function of number of samples (top) as well as the CPU clock time (bottom) for $k = 20$ (left), $k = 25$ (middle-left), $k = 30$ (middle-right) and $k = 35$ (right).

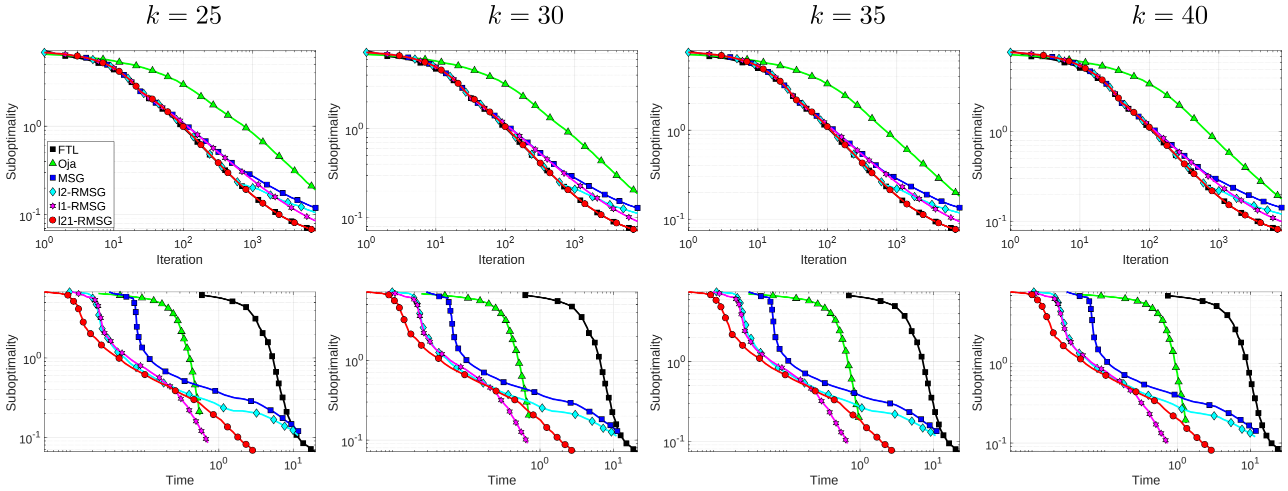

Figure 3: Comparisons of FTL, MSG, ℓ_2-RMSG, ℓ_1-RMSG, $\ell_{2,1}$-RMSG and Oja's algorithm on MNIST dataset, in terms of the suboptimality as a function of number of samples (top) as well as the CPU clock time (bottom) for $k = 25$ (left), $k = 30$ (middle-left), $k = 35$ (middle-right) and $k = 40$ (right).

5. Experimental Results

We provide empirical results for our proposed algorithms ℓ_2-RMSG, ℓ_1-RMSG, and $\ell_{2,1}$-RMSG, compared to vanilla MSG, Oja's algorithm, and Follow The Leader (FTL) algorithm, on both synthetic and real datasets. The synthetic data is drawn from a $d = 100$ dimensional zero-mean multivariate Gaussian distribution with an exponential decay in the spectrum of the covariance matrix. The synthetic consists of $n = 30$K samples, out of which 20K samples are used for training and 5K each for tuning and testing. For comparisons on a real dataset, we choose MNIST which consists of $n = 60$K samples each of size $d = 784$.

The plots in Figures 2 and 3 correspond to the progress in terms of suboptimality in objective as a function of number of samples (top) and the CPU clock time (bottom) for the synthetic dataset and MNIST, respectively. Figure 4 tracks the rank of the iterates for various algorithms. All plots are averaged over 100 runs of the algorithms. The runtime is captured in a controlled setting – each run for every algorithm was on a dedicated identical compute node. The target dimensionality in our experiments is $k \in \{20, 25, 30, 35, 40\}$, however we observed similar behavior for other values of k as well.

For MSG and ℓ_1-RMSG, the learning rate is set to $\frac{\eta_0}{\sqrt{t}}$, and for ℓ_2-RMSG, $\ell_{2,1}$-RMSG and Oja the learning rate was set to $\frac{\eta_0}{t}$ as suggested by theory. We choose η_0 (initial learning rate), λ and μ by tuning[2] each over the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ on held-out data, for $k = 40$. These parameters are then used for all experiments (different values of k). Few remarks are in order.

Iteration complexity. On both the synthetic and the MNIST datasets, we observe that ℓ_2-RMSG and $\ell_{2,1}$-RMSG enjoy faster convergence (better iteration complexity) compared to other MSG variants, as suggested by theory. Surprisingly, Oja's algorithm, despite having a fast $O(\frac{1}{t})$ iteration complexity, is always among the slowest to converge; we remark that similar behavior was noted in previous studies as well (Arora et al., 2012).

Rank of iterates. As can be seen in Figure 4, the rank of the iterates of ℓ_2-RMSG and vanilla MSG quickly increases and hits the maximum while ℓ_1-RMSG and $\ell_{2,1}$-RMSG exhibit good control on the rank of the intermediate iterates. For Oja's algorithm and FTL, by construction, the rank of the iterates is always equal to the desired rank k.

Overall runtime. It can be seen that $\ell_{2,1}$-RMSG and ℓ_1-RMSG consistently outperform other stochastic algorithms. This is interesting because ℓ_2-RMSG has better iteration complexity than ℓ_1-RMSG, but since ℓ_1-RMSG controls the rank of the intermediate iterates, it enjoys a better overall runtime. Recall that each iteration of MSG variants requires $O(dk'^2)$ runtime, where k' is the rank of the current iterate.

Overall analysis: In our experiments, ℓ_1-RMSG and $\ell_{2,1}$-RMSG consistently outperform other stochastic algorithms including vanilla MSG, ℓ_2-RMSG and Oja's algorithm. While ℓ_2-RMSG, $\ell_{2,1}$-RMSG and Oja's algorithm enjoy *optimal* $O(\frac{1}{t})$ iteration complexity, in our experiments, Oja's algorithm is always dominated by ℓ_2-RMSG and $\ell_{2,1}$-RMSG both in iteration complexity as well as the overall runtime. This makes a strong case for MSG variants since Oja's algorithm has optimal computational cost per iteration (linear in input dimension).

6. Discussion

In this paper, we study variants of stochastic gradient descent for a convex relaxation of principal component analysis (PCA), with ℓ_2, ℓ_1, and $\ell_{2,1}$ regularization. We characterize sufficient conditions on the regularization parameters under which an optimum of the regularized problem is also an optimum of the original problem. We show that SGD on the ℓ_2-regularized problem, which we term ℓ_2-RMSG, achieves optimal $O(\frac{1}{\epsilon})$ iteration complexity, whereas SGD on the ℓ_1-regularized problem, which we term ℓ_1-RMSG, provides better control on the rank of the intermediate iterates, which results in smaller computation cost per iteration. We also study $\ell_{2,1}$-RMSG, which leverages both ℓ_1- and ℓ_2-regularization, with the goal of simultaneously improving iteration complexity and computational cost per iteration.

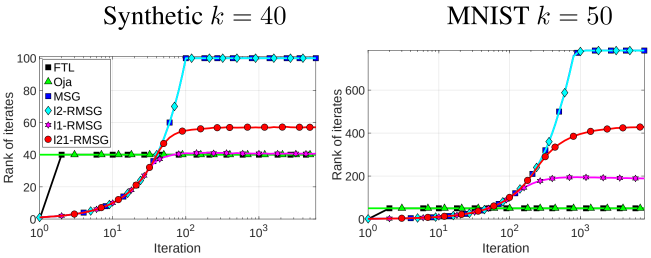

Figure 4: Ranks of iterates of different algorithms discussed in this paper for a synthetic dataset with $k = 40$ (left) and MNIST dataset with $k = 50$ (right). For FTL and Oja, the rank of the iterates is fixed and identical to the desired rank. Rank of iterates of vanilla MSG and ℓ_2-RMSG quickly grows and hits the maximum. Among all variants of MSG, ℓ_1-RMSG has the best control over the rank, while $\ell_{2,1}$-RMSG exhibits a tradeoff between the rank of iterates with sample complexity, as can be seen in Figures 2 and 3.

Our analysis shows that if the learning rate and regularization parameters are chosen appropriately, the expected rank of the iterates of ℓ_1-RMSG is upper bounded by $\tilde{O}(\sqrt{d}/g_k)$. While this results in significant improvement over the per iteration computational cost of MSG (from worst case bound of $O(d^3)$ to $O(d^2)$), it is not yet optimal. In particular, Oja's algorithm enjoys per iteration cost of $O(dk^2)$. Bridging this gap will be the subject for future work.

We provide empirical evidence of our theoretical findings, by comparing several stochastic algorithms on both synthetic and real datasets. Our experiments suggest that ℓ_2- and $\ell_{2,1}$-RMSG are fastest in terms of iteration complexity, while ℓ_1- and $\ell_{2,1}$-RMSG usually enjoy the fastest overall runtime. While our iteration complexity results in Section 2 provide minimax optimal rates for ℓ_2-RMSG and close the gap between iteration complexity of SGD on convex and non-convex programs in Problems 1 and 4, respectively, our analysis fails to provide a certificate for best overall runtime. However, our experiments show that $\ell_{2,1}$-RMSG does benefit from the ℓ_2- and ℓ_1-regularizations, both in the iteration complexity as well as the computational cost per iterate. Providing theoretical guarantees for this empirical finding is another open question we leave to future work.

Another interesting research direction is to revisit issues of statistical and computational efficiency for convex relaxations for related component analysis techniques such as partial least squares (Arora et al., 2016) and canonical correlation analysis (Arora et al., 2017) compared with counterparts based on Oja's algorithm (Ge et al., 2016). Finally, it is natural to consider extensions of the methods proposed here to noisy streaming settings of Marinov et al. (2018).

[2] The performance measure is the PCA objective on the held-out set, after consecutive iterates of the algorithm doesn't improve the objective by a certain amount.

Acknowledgements

This research was supported in part by NSF BIGDATA grant IIS-1546482.

References

Allen-Zhu, Zeyuan and Li, Yuanzhi. First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pp. 487–492. IEEE, 2017.

Arora, Raman, Cotter, Andrew, Livescu, Karen, and Srebro, Nathan. Stochastic optimization for PCA and PLS. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 861–868. IEEE, 2012.

Arora, Raman, Cotter, Andy, and Srebro, Nati. Stochastic optimization of PCA with capped msg. In *Advances in Neural Information Processing Systems*, pp. 1815–1823, 2013.

Arora, Raman, Mianjy, Poorya, and Marinov, Teodor. Stochastic optimization for multiview representation learning using partial least squares. In *International Conference on Machine Learning*, pp. 1786–1794, 2016.

Arora, Raman, Marinov, Teodor Vanislavov, Mianjy, Poorya, and Srebro, Nati. Stochastic approximation for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, pp. 4778–4787, 2017.

Balcan, Maria-Florina, Du, Simon S, Wang, Yining, and Yu, Adams Wei. An improved gap-dependency analysis of the noisy power method. In *29th Annual Conference on Learning Theory*, pp. 284–309, 2016.

Balsubramani, Akshay, Dasgupta, Sanjoy, and Freund, Yoav. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, pp. 3174–3182, 2013.

Ge, Rong, Jin, Chi, Netrapalli, Praneeth, Sidford, Aaron, et al. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *International Conference on Machine Learning*, pp. 2741–2750, 2016.

Jain, Prateek, Jin, Chi, Kakade, Sham M, Netrapalli, Praneeth, and Sidford, Aaron. Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for ojas algorithm. In *29th Annual Conference on Learning Theory*, pp. 1147–1164, 2016.

Marinov, Teodor Vanislavov, Mianjy, Poorya, and Arora, Raman. Streaming principal component analysis in noisy settings. In *International Conference on Machine Learning*, 2018.

Mitliagkas, Ioannis, Caramanis, Constantine, and Jain, Prateek. Memory limited, streaming PCA. In *Advances in Neural Information Processing Systems*, pp. 2886–2894, 2013.

Oja, Erkki. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3): 267–273, 1982.

Rakhlin, Alexander, Shamir, Ohad, Sridharan, Karthik, et al. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning*, 2012.

Shamir, Ohad. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. In *International Conference on Machine Learning*, pp. 248–256, 2016.

Shamir, Ohad and Zhang, Tong. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pp. 71–79, 2013.

Tao, Terence. *Topics in random matrix theory*, volume 132. American Mathematical Society Providence, RI, 2012.

Warmuth, Manfred K and Kuzmin, Dima. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(10), 2008.