Bull. Korean Math. Soc. **0** (0), No. 0, pp. 1–0
https://doi.org/10.4134/BKMS.b170770
pISSN: 1015-8634 / eISSN: 2234-3016

A STOCHASTIC VARIANCE REDUCTION METHOD FOR PCA BY AN EXACT PENALTY APPROACH

Yoon Mo Jung, Jae Hwa Lee, and Sangwoon Yun

Abstract. For principal component analysis (PCA) to efficiently analyze large scale matrices, it is crucial to find a few singular vectors in cheaper computational cost and under lower memory requirement. To compute those in a fast and robust way, we propose a new stochastic method. Especially, we adopt the stochastic variance reduced gradient (SVRG) method [11] to avoid asymptotically slow convergence in stochastic gradient descent methods. For that purpose, we reformulate the PCA problem as a unconstrained optimization problem using a quadratic penalty. In general, increasing the penalty parameter to infinity is needed for the equivalence of the two problems. However, in this case, exact penalization is guaranteed by applying the analysis in [24]. We establish the convergence rate of the proposed method to a stationary point and numerical experiments illustrate the validity and efficiency of the proposed method.

1. Introduction

Principal component analysis (PCA) is a classic tool for data analysis, visualization and dimensionality reduction and has numerous applications in science and engineering [12]. PCA pursues linear combinations of the variables, called principal components, corresponding to directions maximizing variance in the data.

For a given mean-centered data matrix $A = [A_1 \cdots A_n] \in \mathbb{R}^{d\times n}$, where A_i is its i-th column, a PCA algorithm solves the following optimization problem

$$\max_{X\in\mathbb{R}^{d\times r}} \frac{1}{n}\left\|A^\top X\right\|_F^2 \quad \text{subject to} \quad X^\top X = I_r, \tag{1}$$

where d and n are the number of variables and samples respectively, and I_r is the $r \times r$ identity matrix. The task is equivalent to find the r orthonormal eigenvectors associated with the r largest eigenvalues of the $d \times d$ covariance matrix $\frac{1}{n}AA^\top$, or to find the top r left singular vectors of the matrix $\frac{1}{\sqrt{n}}A$

Received September 5, 2017; Revised December 29, 2017; Accepted March 8, 2018.
2010 *Mathematics Subject Classification.* 62H25, 90C30, 15A18.
Key words and phrases. principal component analysis, stochastic variance reduction, exact penalty.

[9]. Since computing an eigendecomposition and a singular value decomposition (SVD) are fundamental problems in matrix computations, they have been extensively studied during decades. Diverse approaches for solving these problems have been proposed and analyzed, and included in various numerical software packages. Further details can be found in [8, 23, 25], etc.

PCA is often obtained by standard algorithms for eigendecompositions or SVDs. A full singular value decomposition may be derived, if the size of matrices is modest [20]. For large scale cases, efficient iterative methods are required due to memory limitation and high computational cost, for example. If matrices are sparse or structured, subspace iteration methods such as Arnoldi method and Lanczos method are available [15, 17]. However, to analyze high dimensional noisy data, a low-dimensional representation is indispensable. In this case, the given data matrices are assumed to be low-rank, and usually dense. For dimensionality reduction, a few leading or dominant eigenvectors or singular vectors are used for such a representation [6].

To efficiently compute a few singular vectors in low-rank large scale matrices, we propose a new stochastic method. To overcome asymptotically slow convergence in stochastic gradient descent methods due to the innate variance, we adopt the stochastic variance reduced gradient (SVRG) method [11]. Nonetheless, SVRG is usually applied to unconstrained problems or constrained problems with special properties on the constraints, for example, Riemannian structure [13, 26, 27]. To apply SVRG effectively, we reformulate the problem (1) as a unconstrained optimization problem using a quadratic penalty. In general, one should increase the penalty parameter to infinity to obtain an optimal solution of the original constrained problem. In our case, with an appropriate penalty parameter, the equivalence between the constrained form and the trace-penalty minimization is guaranteed [24]. It may be treated as an exact penalization. Lastly, we notice that an orthogonalization step such as matrix deflation is not included in the proposed algorithm. Hence the algorithm is low-cost, but it produces a basis for the principal subspace of the desired dimension, instead of the exact singular vectors. That is a major difference from a typical iterative scheme for SVD.

We shortly remark related works. For the problem (1), Shamir proposes stochastic variance reduced gradient methods in the case of $r = 1$ [20] and $r > 1$ [21], respectively. Garber and Hazan [7] use a convex optimization approach, which is originally the inverse power method step in $r = 1$ case. Recently, Allen-Zhu and Li [2] propose an algorithm for $r(> 1)$ singular vectors, which repeatedly performs SVDs for the largest singular vector, based on [7].

The rest of this paper is organized as follows. In Section 2, we give the unconstrained reformulation and explain the exact penalization. Based on it, a new stochastic method for PCA is proposed and its convergence analysis is given in Section 3. In Section 4, we report numerical results and compare with the method in [20]. Finally, concluding remarks are given in Section 5.

2. Unconstrained reformulation and exact penalization

The purpose of this section is deriving a unconstrained optimization problem using a quadratic penalty as a preliminary step to apply SVRG. In addition, we obtain an exact penalization by applying the trace-penalty minimization approach in [24].

By noticing $\mathrm{tr}(X^\top AA^\top X) = \|A^\top X\|_F^2$, the PCA problem (1) can be reformulated as a minimization problem

$$\min_{X\in\mathbb{R}^{d\times r}} \mathrm{tr}\left(X^\top\left(\nu I_d - \frac{1}{n}AA^\top\right)X\right) \quad \text{subject to} \quad X^\top X = I_r, \tag{2}$$

where $\nu > 0$ is a constant to assure the positive definiteness of the matrix $\nu I_d - \frac{1}{n}AA^\top$. Its unconstrained reformulation using the quadratic penalty is

$$\min_{X\in\mathbb{R}^{d\times r}} F_\mu(X) := f(X) + \frac{\mu}{4}\|X^\top X - I_r\|_F^2, \tag{3}$$

where $f(X) = \frac{1}{2}\mathrm{tr}(X^\top(\nu I_d - \frac{1}{n}AA^\top)X)$ and $\mu > 0$ is a penalty parameter.

In general, for the equivalence of those two problems, $\mu \to \infty$ may be required. However, in the case of the problems (2) and (3), if a proper μ is chosen, then they are equivalent. More specifically, the solutions of these two problems span the same eigenspace by Theorem 2.1 in [24]. For completeness, we restate this theorem for the problems (2) and (3).

Theorem 2.1. *The problem* (2) *is equivalent to the problem* (3) *if and only if*

$$\mu > \nu - \lambda_r, \tag{4}$$

where λ_r *is the* r*-th largest eigenvalue of* $\frac{1}{n}AA^\top$.

We can easily see that the optimality condition of the problem (3) implies that of the problem (2). If an orthonormal basis of the range space of X is denoted by $Y(X) \in \mathbb{R}^{d\times r}$ and

$$\begin{aligned} R(X) &= \left(\nu I_d - \frac{1}{n}AA^\top\right)Y(X) - Y(X)\left(Y(X)^\top\left(\nu I_d - \frac{1}{n}AA^\top\right)Y(X)\right) \\ &= -\frac{1}{n}(AA^\top Y(X) - Y(X)(Y(X)^\top AA^\top Y(X))), \end{aligned}$$

it is easily shown that $R(X) = 0$ if and only if $Y(X)$ is a KKT point of the problems (1) and (2). In the case of the PCA problem (2) and its penalized model (3), we have

$$\|R(X)\|_F \le \sigma_{\min}^{-1}(X)\|\nabla F_\mu(X)\|_F,$$

where X is a rank-r matrix and $\sigma_{\min}(X)$ is the smallest singular value of X. Hence, if $\|\nabla F_\mu(X)\|_F \le \epsilon$ for sufficiently small constant $\epsilon > 0$, we can obtain an approximate solution of the problem (2). For details, see [10, 24].

3. A stochastic variance reduced method for penalized problems

In this section, we propose a new stochastic method for PCA (1) using the unconstrained reformulation (3) derived in Section 2. For this purpose, we express the problem (3) as a sum involving a rank-one matrix $A_iA_i^\top$ or each data A_i. Indeed, since

$$f(X) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\mathrm{tr}(X^\top(\nu I_d - A_iA_i^\top)X) := \frac{1}{n}\sum_{i=1}^{n} f_i(X),$$

we have the new formulation of the problem

$$\min_{X\in\mathbb{R}^{d\times r}} \quad \frac{1}{n}\sum_{i=1}^{n}\left(f_i(X) + \frac{\mu}{4}\|X^\top X - I_r\|_F^2\right). \tag{5}$$

In classical stochastic gradient methods [5], only one component function is randomly selected and its gradient is evaluated to approximate the full gradient at each iteration. However, stochastic methods suffer a sublinear convergence in expectation, due to the variance of random sampling. Thus, diverse approaches such as SAG [18], SDCA [19], SVRG [11] are proposed to reduce the variance. For the problem (5), we adopt the stochastic variance reduced gradient (SVRG) method [11], since it reduces the variance explicitly. Indeed, the SVRG method has been widely used due to its cheaper computational cost and lower memory requirements for gradients of component functions than some other stochastic methods [18,19]. For SVRG applied to nonconvex objective functions, we refer to [1,3,16].

By applying SVRG to the problem (5), we have the following update at the k-th iteration of s-th epoch:

$$\begin{aligned} V_k = {} & \nabla f_{i_k}(X_k) + \mu X_k(X_k^\top X_k - I_r) \\ & - (\nabla f_{i_k}(\tilde{X}_s) + \mu\tilde{X}_s(\tilde{X}_s^\top\tilde{X}_s - I_r)) + G_s, \end{aligned} \tag{6}$$

$$X_{k+1} = X_k - \eta_s V_k, \tag{7}$$

where $\eta_s > 0$ is a stepsize, $\tilde{X}_s$ is the output after one pass over the data, which is used for the next epoch, and $G_s = \nabla F_\mu(\tilde{X}_s)$ is the full gradient computed at $\tilde{X}_s$. To accelerate convergence, we use a variant of stochastic Barzilai-Borwein (BB) stepsize, started with an initial stepsize η_0 [4, 22]. We call it the stochastic variance reduced gradient method for PCA (SVRG-PCA) and describe the algorithmic framework in Algorithm 1.

Before establishing a convergence theorem for our algorithm, we notice that the problem (5) is nonconvex due to the penalty term $\|X^\top X - I_r\|_F^2$. Thus, we consider a general nonconvex problem

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} g_i(x), \tag{8}$$

and borrow the idea of Allen-Zhu and Hazan [1] and Reddi et al. [16]; they show global convergence results on the SVRG method and its variants under the following assumption of L-smoothness of component functions g_i.

Algorithm 1 SVRG-PCA

Given: update frequency K, initial point $\tilde{X}_0$, constant ν, penalty parameter μ, initial stepsize η_0 (only used in the first epoch)
for $s = 0, \dots, S-1$ **do**
  $X_0 = \tilde{X}_s$
  $G_s = \nabla F_\mu(\tilde{X}_s)$
  if $s > 0$ **then**
    $\eta_s = \frac{\|\tilde{X}_s - \tilde{X}_{s-1}\|_F^2}{\mathrm{tr}((\tilde{X}_s - \tilde{X}_{s-1})^\top (G_s - G_{s-1}))}$
  end if
  for $k = 0, \dots, K-1$ **do**
    Randomly pick $i_k \in \{1, \dots, n\}$
    Compute V_k
    $X_{k+1} = X_k - \eta_s V_k$
  end for
  $\tilde{X}_{s+1} = X_K$
end for

Assumption 1. $\nabla g_i(x)$ is L-Lipschitz continuous (also g_i is called L-smooth), that is,

$$\|\nabla g_i(x) - \nabla g_i(y)\| \le L \|x - y\| \quad \text{for all } x,\, y \in \mathbb{R}^d. \tag{9}$$

Since each $f_i(X)$ is a quadratic function, its gradient is L-Lipschitz continuous. However, the penalty term $\|X^\top X - I_r\|_F^2$ is not L-smooth on $\mathbb{R}^{d\times r}$. Thus, the analyses in [1] and [16] are not directly applicable. We bypass the difficulty using the local L-smoothness of the penalty term and this property enables us to apply the idea of Reddi et al. [16]. Note that $\mathbb{E}[V_k] = \nabla F_\mu(X_k)$. To prove convergence theorem, we need the following lemma.

Lemma 3.1. *For $c_k, c_{k+1}, \beta_k > 0$, suppose we have*

$$c_k = c_{k+1}(1 + \eta_s\beta_k + 2\eta_s^2 L^2) + \eta_s^2 L^3,$$

and let η_s, β_k and c_{k+1} be chosen as

$$\Gamma_k = \left(\eta_s - \frac{c_{k+1}\eta_s}{\beta_k} - \eta_s^2 L - 2c_{k+1}\eta_s^2\right) > 0. \tag{10}$$

Suppose that the iterate X_k in the k-th iteration of s-th epoch generated by Algorithm 1 belongs to an open convex set D on which $\|X^\top X - I_r\|_F^2$ is L-smooth. Then X_k satisfies the bound

$$\mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \le \frac{R_k - R_{k+1}}{\Gamma_k},$$

where $R_k = \mathbb{E}[f(X_k) + c_k\|X_k - \tilde{X}_s\|_F^2]$ *for* $0 \le s \le S-1$.

Proof. Since $F_\mu(X)$ is L-smooth on D and (7), we have

$$\begin{aligned}\mathbb{E}[F_\mu(X_{k+1})] &\le \mathbb{E}\left[F_\mu(X_k) + \langle \nabla F_\mu(X_k), X_{k+1} - X_k\rangle + \frac{L}{2}\|X_{k+1} - X_k\|_F^2\right] \\ &= \mathbb{E}\left[F_\mu(X_k) - \eta_s\|\nabla F_\mu(X_k)\|_F^2 + \frac{L}{2}\eta_s^2\|V_k\|_F^2\right], \qquad (11)\end{aligned}$$

where $\langle\ ,\ \rangle$ is the inner product defined by $\langle X, Y\rangle = \mathrm{tr}(X^\top Y)$. And we can bound $\mathbb{E}[\|X_{k+1} - \tilde{X}_s\|_F^2]$ from

$$\begin{aligned}\mathbb{E}[\|X_{k+1} - \tilde{X}_s\|_F^2] &= \mathbb{E}[\|X_{k+1} - X_k + X_k - \tilde{X}_s\|_F^2] \\ &= \mathbb{E}[\|X_{k+1} - X_k\|_F^2 + \|X_k - \tilde{X}_s\|_F^2 \\ &\quad + 2\langle X_{k+1} - X_k, X_k - \tilde{X}_s\rangle] \\ &= \mathbb{E}[\eta_s^2\|V_k\|_F^2 + \|X_k - \tilde{X}_s\|_F^2 - 2\eta_s\langle V_k, X_k - \tilde{X}_s\rangle] \\ &= \eta_s^2\mathbb{E}[\|V_k\|_F^2] + \mathbb{E}[\|X_k - \tilde{X}_s\|_F^2] \\ &\quad - 2\eta_s\mathbb{E}[\langle \nabla F_\mu(X_k), X_k - \tilde{X}_s\rangle] \\ &\le \eta_s^2\mathbb{E}[\|V_k\|_F^2] + \mathbb{E}[\|X_k - \tilde{X}_s\|_F^2] \\ &\quad + 2\eta_s\mathbb{E}[\|\nabla F_\mu(X_k)\|_F\|X_k - \tilde{X}_s\|_F] \\ &\le \eta_s^2\mathbb{E}[\|V_k\|_F^2] + \mathbb{E}[\|X_k - \tilde{X}_s\|_F^2] \\ &\quad + 2\eta_s\mathbb{E}\left[\frac{1}{2\beta_k}\|\nabla F_\mu(X_k)\|_F^2 + \frac{1}{2}\beta_k\|X_k - \tilde{X}_s\|_F^2\right], \qquad (12)\end{aligned}$$

where the first and second inequalities follows from Cauchy-Schwarz and Young's inequality, respectively. For $\mathbb{E}[\|V_k\|_F^2]$, we have

$$\begin{aligned}\mathbb{E}[\|V_k\|_F^2] &= \mathbb{E}[\|V_k - \nabla F_\mu(X_k) + \nabla F_\mu(X_k)\|_F^2] \\ &\le 2\mathbb{E}[\|(\nabla f_{i_k}(X_k) - \nabla f_{i_k}(\tilde{X}_s)) - (\nabla f(X_k) - \nabla f(\tilde{X}_s))\|_F^2] \\ &\quad + 2\mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \\ &\le 2\mathbb{E}[\|\nabla f_{i_k}(X_k) - \nabla f_{i_k}(\tilde{X}_s)\|_F^2] + 2\mathbb{E}[\nabla F_\mu(X_k)\|_F^2] \\ &\le 2L^2\mathbb{E}[\|X_k - \tilde{X}_s\|_F^2] + 2\mathbb{E}[\nabla F_\mu(X_k)\|_F^2], \qquad (13)\end{aligned}$$

where the first inequality follows from $\|A+B\|_F^2 \le 2\|A\|_F^2 + 2\|B\|_F^2$, the second inequality is from $\mathbb{E}[\|Y - \mathbb{E}[Y]\|_F^2] = \mathbb{E}[\|Y\|_F^2] - \|\mathbb{E}[Y]\|_F^2 \le \mathbb{E}[\|Y\|_F^2]$ for a random matrix variable Y, and the last inequality is from L-smoothness of f_i. Now we define

$$R_k = \mathbb{E}[F_\mu(X_k) + c_k\|X_k - \tilde{X}_s\|_F^2]. \qquad (14)$$

Then using the inequality (11) and (12), we have

$$R_{k+1} = \mathbb{E}[F_\mu(X_{k+1}) + c_{k+1}\|X_{k+1} - \tilde{X}_s\|_F^2]$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[F_\mu(X_k) - \eta_s\|\nabla F_\mu(X_k)\|_F^2 + \frac{L}{2}\eta_s^2\|V_k\|_F^2\right] \\
&\quad + \mathbb{E}[c_{k+1}\eta_s^2\|V_k\|_F^2 + c_{k+1}\|X_k - \tilde{X}_s\|_F^2] \\
&\quad + 2c_{k+1}\eta_s\mathbb{E}\left[\frac{1}{2\beta_k}\|\nabla F_\mu(X_k)\|_F^2 + \frac{1}{2}\beta_k\|X_k - \tilde{X}_s\|_F^2\right] \\
&\leq \mathbb{E}\left[F_\mu(X_k) - \left(\eta_s - \frac{c_{k+1}\eta_s}{\beta_k}\right)\|\nabla F_\mu(X_k)\|_F^2\right] \\
&\quad + \left(\frac{L}{2}\eta_s^2 + c_{k+1}\eta_s^2\right)\mathbb{E}[\|V_k\|_F^2] \\
&\quad + \left(c_{k+1} + c_{k+1}\eta_s\beta_k\right)\mathbb{E}[\|X_k - \tilde{X}_s\|_F^2].
\end{aligned}
$$

Using the inequality (13) and the definition (14), we have

$$
\begin{aligned}
R_{k+1} &\leq \mathbb{E}[F_\mu(X_k)] \\
&\quad - \left(\eta_s - \frac{c_{k+1}\eta_s}{\beta_k} - \eta_s^2 L - 2c_{k+1}\eta_s^2\right)\mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \\
&\quad + [c_{k+1}(1 + \eta_s\beta_k + 2\eta_s^2L^2) + \eta_s^2L^3]\mathbb{E}[\|X_k - \tilde{X}_s\|_F^2] \\
&\leq R_k - \left(\eta_s - \frac{c_{k+1}\eta_s}{\beta_k} - \eta_s^2 L - 2c_{k+1}\eta_s^2\right)\mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2].
\end{aligned}
$$

Hence, using the definition of Γ_k (10), we can obtain the conclusion. □

We establish the convergence rate of Algorithm 1 to a stationary point in the following theorem. Due to the stochastic nature of the algorithm, we select an iterate uniform randomly as an output for the theorem, instead of the last iterate.

Theorem 3.2. *Suppose that the condition of Lemma 3.1 is satisfied. Let* $c_K = 0, \eta_k = \eta > 0, \beta_k = \beta > 0$, *and* $c_k = c_{k+1}(1 + \eta\beta + 2\eta^2L^2) + \eta^2L^3$ *such that* $\Gamma_k > 0$ *for* $0 \leq k \leq K-1$. *Define the quantity* $\gamma_n := \min_k \Gamma_k$. *Further, let* T *be a multiple of* K. *Then, for the output* X_a *of Algorithm 1, we have*

$$
\mathbb{E}[\|\nabla F_\mu(X_a)\|_F^2] \leq \frac{F_\mu(\tilde{X}_0) - F_\mu(X^*)}{T\gamma_n},
$$

where X^* *is an optimal solution to* (5).

Proof. In the s-th epoch, by using Lemma 3.1, we have

$$
\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \leq \frac{R_0 - R_K}{\gamma_n}.
$$

Since $R_0 = \mathbb{E}[F_\mu(\tilde{X}_s)]$ and $R_K = \mathbb{E}[F_\mu(\tilde{X}_{s+1})] = \mathbb{E}[F_\mu(X_K)]$, the above inequality implies

$$\sum_{k=0}^{K-1} \mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \leq \frac{\mathbb{E}[F_\mu(\tilde{X}_s) - F_\mu(\tilde{X}_{s+1})]}{\gamma_n}.$$

Summing over all epochs, we get

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{k=0}^{K-1} \mathbb{E}[\|\nabla F_\mu(X_k)\|_F^2] \leq \frac{F_\mu(\tilde{X}_0) - F(X^*)}{T\gamma_n}.$$

Using this inequality, we obtain the conclusion. □

4. Numerical results

In this section, we report numerical results for our algorithm. We implement experiments in MATLAB R2015b and perform on a PC Intel CoreTM i5 (3.50 GHz) processor in Windows 10 operating system.

For the positive definiteness of the matrix $\nu I_d - \frac{1}{n}AA^\top$, we choose $\nu = \frac{c_1}{n}\|A\|_F^2$ and $\mu = c_2\nu$, accordingly. We select $c_1 = 2$, $c_2 = 0.2$ by a few preliminary experiments. The initial $\tilde{X}_0$ is randomly generated and we set the maximum number of epochs to 100 and the update frequency $K = \frac{n}{50}$ for $1000 \leq n < 10000$, $K = \frac{n}{100}$ for $n \geq 10000$, respectively. The stopping criterion is chosen as

$$\|\nabla F_\mu(\tilde{X}_s)\|_F \leq 10^{-8}.$$

We test four types of data matrices; Gaussian random matrices, low-rank dense matrices with Gaussian noise, synthetic random datasets with various eigengaps used in [20], and the well-known MNIST data matrix [14]. For Gaussian random matrices, we test the following three cases; i) $n = 1000$, $d = 100$, and $r = 2, 5, 10$, ii) $n = 5000$, $d = 500$, and $r = 10, 20, 50$, iii) $n = 10000$, $d = 1000$, and $r = 20, 40, 60$. Here, d, n, and r denote the number of variables, samples, and singular vectors, respectively. For low-rank dense matrices with Gaussian noise, i) $n = 1000$, $d = 100$ with rank 10, and $r = 2, 5, 10$ ii) $n = 5000$, $d = 500$, with rank 50 and $r = 10, 20, 50$, iii) $n = 10000$, $d = 1000$, with rank 60, and $r = 20, 40, 60$. The synthetic datasets in [20] are of the form $A = UDV^\top$, where $D = \mathrm{diag}(1, 1-\lambda, 1-1.1\lambda, 1-1.2\lambda, 1-1.3\lambda, 1-1.4\lambda, q_1, q_2, \ldots)$, $q_i = |g_i|/d$, g_i are randomly chosen small quantities, and U and V are random orthogonal matrices. We test the following three cases with the eigengaps $\lambda = 0.16, 0.05, 0.016, 0.005, 0.0016$; i) $n = 1000$, $d = 100$, and $r = 3, 6$, ii) $n = 5000$, $d = 500$, and $r = 3, 6$, iii) $n = 10000$, $d = 1000$, and $r = 3, 6$. The size of the MNIST data matrix is 784×70000, and we test two cases $r = 3, 6$.

We compare SVRG-PCA with Shamir's VR-PCA [20]. We first run the proposed algorithm and save the number of epochs and CPU time. Then, we execute VR-PCA algorithm for the same number of epochs. Since there is a very little time difference between the original VR-PCA [20] and the variant [21], we only report the results of [20]. Since the standard algorithms for PCA

TABLE 1. Numerical results for Gaussian random matrices

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	2	0.0164	10.9030	0.0219	9.9194	12.8654
		5	0.0171	17.9460	0.0222	15.4513	19.6773
		10	0.0179	27.0457	0.0270	22.7817	28.5917
5000	500	10	2.1963	52.0787	2.6444	43.6430	56.4699
		20	2.1803	73.1929	2.5954	60.8326	77.9029
		50	2.2834	117.3563	3.5554	97.9688	122.5267
10000	1000	20	9.6512	101.7981	11.2504	84.6626	109.5969
		40	10.4210	142.9740	12.6819	118.5387	151.8802
		60	11.7194	175.7425	14.4414	145.5593	184.9345

TABLE 2. Numerical results for low-rank dense matrices with Gaussian noise

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	2	0.0223	37.6153	0.0250	25.8904	69.4663
		5	0.0261	55.4922	0.0346	37.2277	74.1922
		10	0.0257	82.1014	0.0389	47.4114	82.1015
5000	500	10	1.6907	149.6466	1.9582	74.7004	303.2602
		20	1.8943	210.7750	2.2929	89.9494	317.2930
		50	2.4835	349.7847	3.6714	141.8820	349.7848
10000	1000	20	11.7270	385.0424	13.4533	133.8535	610.9023
		40	12.7530	530.6809	14.7403	162.1582	628.0962
		60	14.7440	640.7248	17.1421	193.4564	640.7249

provide an orthonormal basis, we apply the Rayleigh-Ritz (RR) procedure [24] on the final iterate $\tilde{X}_s$. This RR procedure also gives the approximated singular values. The time required for this process is tiny compared with main loop and also included in CPU time.

The results are reported in Tables 1–8. In each table, 'time' denotes the CPU time in seconds and $\|A^\top X\|_F$ is the square root of the objective value of the PCA problem (1) with the orthogonal matrix X. 'sslev' denotes the value of square root of the sum of the r largest eigenvalues of $AA^\top$. For all tests, 10 runs are executed. Among them, the average values are reported in each table. The closeness of quantity $\|A^\top X\|_F$ to 'sslev' measures the quality of algorithms; ideally, they should be equal. The results of SVRG-PCA is better than those of VR-PCA. Furthermore, to run the same number of epochs, the proposed algorithm is faster than VR-PCA.

TABLE 3. Numerical results for synthetic random datasets with eigengap 0.16 in [20]

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	3	0.0185	1.4812	0.0221	0.8181	1.5442
		6	0.0257	2.0605	0.0314	1.1472	2.0656
5000	500	3	1.6977	1.4651	1.9499	0.8907	1.5442
		6	2.1001	2.0654	2.7892	1.2213	2.0656
10000	1000	3	10.0062	1.4769	11.6384	0.4103	1.5442
		6	12.0435	2.0653	13.9382	0.6836	2.0656

TABLE 4. Numerical results for synthetic random datasets with eigengap 0.05 in [20]

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	3	0.0217	1.6448	0.0259	0.8785	1.6720
		6	0.0227	2.3245	0.0288	1.2401	2.3277
5000	500	3	2.2194	1.6489	2.3791	1.0147	1.6720
		6	3.8521	2.3273	3.7237	1.2647	2.3277
10000	1000	3	10.5106	1.6495	12.1033	0.4689	1.6720
		6	9.3105	2.3276	10.5586	0.5947	2.3277

TABLE 5. Numerical results for synthetic random datasets with eigengap 0.016 in [20]

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	3	0.0186	1.7031	0.0226	0.9352	1.7127
		6	0.0223	2.4063	0.0277	1.3011	2.4104
5000	500	3	1.9333	1.7042	2.1981	0.9232	1.7127
		6	2.1334	2.4102	2.4446	1.3882	2.4104
10000	1000	3	9.6348	1.7035	10.9962	0.4862	1.7127
		6	9.5765	2.4102	10.9944	0.6804	2.4104

5. Concluding remarks

In this paper, we adopt a stochastic variance reduced gradient method for solving principal component analysis (PCA) problems. To apply SVRG properly, we reformulate the PCA problem as a penalized unconstrained optimization problem. Exact penalization is guaranteed by applying the analysis in [24] and we show the convergence rate of the proposed algorithm to a stationary point. Numerical experiments and comparison with the recently proposed

TABLE 6. Numerical results for synthetic random datasets with eigengap 0.005 in [20]

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	3	0.0197	1.7228	0.0249	1.0134	1.7260
		6	0.0236	2.4365	0.0285	1.2740	2.4372
5000	500	3	2.0664	1.7228	2.2251	0.9540	1.7260
		6	2.3973	2.4369	2.4946	1.4518	2.4372
10000	1000	3	9.6094	1.7234	11.0638	0.4491	1.7260
		6	9.5354	2.4369	10.8956	0.6193	2.4372

TABLE 7. Numerical results for synthetic random datasets with eigengap 0.0016 in [20]

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
1000	100	3	0.0222	1.7279	0.0232	0.9350	1.7301
		6	0.0231	2.4260	0.0280	1.3210	2.4456
5000	500	3	2.0182	1.7289	2.3746	1.0017	1.7301
		6	2.2612	2.4453	2.6705	1.4754	2.4456
10000	1000	3	11.0240	1.7291	11.9103	0.5085	1.7301
		6	10.0172	2.4455	11.5235	0.6992	2.4456

TABLE 8. Numerical results for MNIST dataset

n	d	r	SVRG-PCA		VR-PCA		sslev
			time	$\|A^\top X\|_F$	time	$\|A^\top X\|_F$	
70000	784	3	64.0395	1712.0840	69.1427	1657.4024	1803.8504
		6	65.4350	1851.7406	70.5407	1677.0609	1958.1236

stochastic method VR-PCA [20] illustrate the effectiveness of the proposed method.

Acknowledgement. The authors would like to thank Professor Bo Jiang and another anonymous referee for their kind advice and modifications. This work was supported by the National Research Foundation of Korea (NRF) NRF-2016R1A5A1008055. The first author was supported by the National Research Foundation of Korea (NRF) NRF-2016R1D1A1B03931337. The third author was supported by the National Research Foundation of Korea (NRF) NRF-2016R1D1A1B03934371.

References

[1] Z. Allen-Zhu and E. Hazan, *Variance Reduction for Faster Non-Convex Optimization*, Preprint arXiv:1603.05643, 2016.

[2] Z. Allen-Zhu and Y. Li, *LazySVD: Even Faster SVD Decomposition Yet Without Agonizing Pain*, Preprint arXiv:1607.03463v2, 2017.
[3] Z. Allen-Zhu and Y. Yuan, *Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives*, Preprint arXiv:1506.01972v3, 2016.
[4] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal. **8** (1988), no. 1, 141–148.
[5] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization Methods for Large-Scale Machine Learning*, Preprint arXiv:1606.04838v1, 2016.
[6] J. P. Cunningham and Z. Ghahramani, *Linear dimensionality reduction: survey, insights, and generalizations*, J. Mach. Learn. Res. **16** (2015), 2859–2900.
[7] D. Garber and E. Hazan, *Fast and Simple PCA via Convex Optimization*, Preprint arXiv:1509.05647v4, 2015.
[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, fourth edition, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013.
[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, second edition, Cambridge University Press, Cambridge, 2013.
[10] B. Jiang, C. Cui, and Y.-H. Dai, *Unconstrained optimization models for computing several extreme eigenpairs of real symmetric matrices*, Pac. J. Optim. **10** (2014), no. 1, 53–71.
[11] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, Advances in Neural Information Processing Systems (2013), 315–323.
[12] I. T. Jolliffe, *Principal Component Analysis*, second edition, Springer Series in Statistics, Springer-Verlag, New York, 2002.
[13] H. Kasai, H. Sato, and B. Mishra, *Riemannian stochastic variance reduced gradient on Grassmann manifold*, Preprint arXiv:1605.07367v3, 2017.
[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, **86** (1998), no. 11, 2278–2324.
[15] X. Liu, Z. Wen, and Y. Zhang, *Limited memory block Krylov subspace optimization for computing dominant singular value decompositions*, SIAM J. Sci. Comput. **35** (2013), no. 3, A1641–A1668.
[16] S. J. Reddi, A. Hefny, S. Sra, and B. Poczos, *Stochastic Variance Reduction for Nonconvex Optimization*, Preprint arXiv:1603.06160v2, 2016.
[17] Y. Saad, *Numerical methods for large eigenvalue problems*, revised edition of the 1992 original, Classics in Applied Mathematics, **66**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011.
[18] M. Schmidt, N. Le Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Math. Program. **162** (2017), no. 1-2, Ser. A, 83–112.
[19] S. Shalev-Shwartz and T. Zhang, *Stochastic dual coordinate ascent methods for regularized loss minimization*, J. Mach. Learn. Res. **14** (2013), 567–599.
[20] O. Shamir, *A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rate*, In The 32nd International Conference on Machine Learning (ICML 2015), 2015.
[21] ———, *Fast stochastic algorithms for SVD and PCA*: *convergence properties and convexity*, Preprint arXiv:1507.08788v1, 2015.
[22] C. Tan, S. Ma, Y. -H. Dai, and Y. Qian, *Barzilai-Borwein step size for stochastic gradient descent*, Preprint arXiv:1605.04131v2, 2016.
[23] D. S. Watkins, *The Matrix Eigenvalue Problem*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.
[24] Z. Wen, C. Yang, X. Liu, and Y. Zhang, *Trace-penalty minimization for large-scale eigenspace computation*, J. Sci. Comput. **66** (2016), no. 3, 1175–1203.
[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Monographs on Numerical Analysis, The Clarendon Press, Oxford University Press, New York, 1988.

[26] Z. Xu and Y. Ke, *Stochastic variance reduced Riemannian eigensolver*, Preprint arXiv:1605.08233v2, 2016.
[27] H. Zhang, S. J. Reddi, and S. Sra, *Fast stochastic optimization on Riemannian manifolds*, Preprint arXiv:1605.07147v2, 2017.

Yoon Mo Jung
Department of Mathematics
Sungkyunkwan University
Suwon 16419, Korea
Email address: yoonmojung@skku.edu

Jae Hwa Lee
Applied Algebra and Optimization Research Center
Sungkyunkwan University
Suwon 16419, Korea
Email address: jhlee2chn@skku.edu

Sangwoon Yun
Department of Mathematics Education
Sungkyunkwan University
Seoul 03063, Korea
Email address: yswmathedu@skku.edu