Matrix Eigen-decomposition via Doubly Stochastic Riemannian Optimization

Zhiqiang Xu XUZQ@I2R.A-STAR.EDU.SG
Peilin Zhao ZHAOP@I2R.A-STAR.EDU.SG
Jianneng Cao CAOJN@I2R.A-STAR.EDU.SG
Xiaoli Li XLLI@I2R.A-STAR.EDU.SG
Institute for Infocomm Research, A*STAR, Singapore

Abstract

Matrix eigen-decomposition is a classic and long-standing problem that plays a fundamental role in scientific computing and machine learning. Despite some existing algorithms for this inherently non-convex problem, the study remains inadequate for the need of large data nowadays. To address this gap, we propose a **D**oubly **S**tochastic **R**iemannian **G**radient **EIG**en**S**olver, DSRG-EIGS, where the double stochasticity comes from the generalization of the stochastic Euclidean gradient ascent and the stochastic Euclidean coordinate ascent to Riemannian manifolds. As a result, it induces a greatly reduced complexity per iteration, enables the algorithm to completely avoid the matrix inversion, and consequently makes it well-suited to large-scale applications. We theoretically analyze its convergence properties and empirically validate it on real-world datasets. Encouraging experimental results demonstrate its advantages over the deterministic counterpart.

1. Introduction

Matrix eigen-decomposition, aiming at a group of top eigenvectors of a given matrix (Golub & Van Loan, 1996), has found widespread applications in many areas of scientific and engineering computing, e.g., numerical computation (Press et al., 2007) and structure analysis (Torbjorn Ringertz, 1997)). Particularly, it plays a fundamental role in many machine learning tasks, such as spectral clustering (Ng et al., 2002), dimensionality reduction (Jolliffe, 2002), and kernel approximation (Drineas & Mahoney, 2005), etc. Despite the great importance of this problem, existing solutions, i.e., eigensolvers, have been relatively lacking. Among them, the power method (Golub & Van Loan, 1996) and the (block) Lanczos algorithm (Parlett, 1998) belong to well-known eigensolvers, while randomized SVD (Halko et al., 2011) and online learning of eigenvectors (Garber et al., 2015) are recently proposed. In addition, matrix eigendecompostion can be formulated as a quadratically constrained quadratic program, and thus can be addressed from the optimization perspective, for example, the trace penalty minimization (Wen et al., 2013). Notably, its non-convex constraint set constitutes a Riemannian manifold, or more precisely, Stiefel manifold, which turns it into a Riemannian optimization problem that can be tackled by the methods of optimization on manifolds (Edelman et al., 1999; Absil et al., 2008; Wen & Yin, 2013). However, most of existing eigensolvers belong to batch learning, i.e., using the entire dataset at each update step, and thus are not suitable to large-scale matrices, especially those unable to completely fit into memory. To address this issue, we usually could resort to stochastic optimization, which enables the algorithm to work through access to only a subset of the data each time. And stochastic algorithms often converge faster than their batch counterparts even if no memory issue arises.

To overcome the limitations of existing batch learning eigensolvers, we propose a doubly stochastic Riemannian gradient method to obtain the **DSRG-EIGS** algorithm, a new eigensolver. The method simultaneously generalizes the stochastic gradient ascent (SGA) and the stochastic coordinate ascent (SCA) (Nesterov, 2012) from the Euclidean space to the Riemannian space, and arrives at a combination of their Riemannian counterparts: stochastic Riemannian gradient ascent (SRGA) and stochastic Riemannian coordinate ascent (SRCA). Specifically, SRGA works by sampling data sub-matrices, while SRCA proceeds by sampling column blocks of Riemannian gradient coordinates in our problem. Both methods keep iterates remaining on the manifold and stochastic Riemannian gradients staying in the tangent space during iterations. They greatly reduce the complexity per iteration of the algorithm, especially for

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

dense matrices. Meanwhile, the algorithm becomes able to completely avoid the matrix inversion required in its deterministic version, and thus can work effectively in the case of desiring a large number of eigenvectors. Furthermore, we provide a progressive analysis on the theoretical convergence properties of DSRG-EIGS, which shows the convergence of the algorithm to global solutions at a sub-linear rate in expectation and that the algorithm is able to take advantage of importance sampling (Zhao & Zhang, 2014) to improve the convergence rate.

The rest of the paper is organized as follows. In Section 2, we review some basics of matrix eigen-decomposition and Riemannian optimization. We present our doubly stochastic Riemannian gradient eigensolver, abbreviated as DSRG-EIGS in Section 3, followed by the progressive theoretical analysis in Section 4. Experimental results are shown in Section 5. Section 6 discusses related work. Finally, Section 7 concludes the paper.

2. Preliminaries

2.1. Matrix Eigen-decomposition

The eigen-decomposition of a symmetric[1] matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ says that $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$ ($\mathbf{u}_j$ represents the j^{th} column of $\mathbf{U}$) is an orthogonal matrix, i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I}$ ($\mathbf{I}$ represents the identity matrix of appropriate size), $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ is a diagonal matrix, and $\mathbf{u}_j$ is called the eigenvector corresponding to the eigenvalue λ_j, i.e., $\mathbf{A}\mathbf{u}_j = \lambda_j\mathbf{u}_j$. For the convenience in the sequel, we assume that $\lambda_1 \geq \cdots \geq \lambda_n$ and define $\mathbf{V} \triangleq [\mathbf{u}_1, \cdots, \mathbf{u}_q]$ and $\mathbf{V}_\perp \triangleq [\mathbf{u}_{q+1}, \cdots, \mathbf{u}_n]$, $\mathbf{\Sigma} \triangleq \mathrm{diag}(\lambda_1, \cdots, \lambda_q)$ and $\mathbf{\Sigma}_\perp \triangleq \mathrm{diag}(\lambda_{q+1}, \cdots, \lambda_n)$, where q is the number of top eigenvectors to be sought.

From the point of view of optimization, in practice, matrix eigen-decomposition can be defined by the following nonconvex quadratically constrained quadratic program:

$$\max_{\mathbf{X}\in\mathbb{R}^{n\times q}:\mathbf{X}^\top\mathbf{X}=\mathbf{I}} (1/2)\mathrm{tr}(\mathbf{X}^\top\mathbf{A}\mathbf{X}), \tag{1}$$

where $q < n$ and $\mathrm{tr}(\cdot)$ represents the trace of a square matrix, i.e., sum of its diagonal entries. It can be easily verified that $\mathbf{X} = \mathbf{V}$ maximizes the trace value at $(1/2)\sum_{i=1}^q \lambda_i$.

2.2. Riemannian Gradient

Given a Riemannian manifold (Lee, 2012) $\mathcal{M}$, its tangent space at a point $\mathbf{X} \in \mathcal{M}$, denoted as $T_\mathbf{X}\mathcal{M}$, is a Euclidean space that locally linearizes $\mathcal{M}$ around $\mathbf{X}$. Analogous to the Euclidean case, one iterate of the first-order optimization on $\mathcal{M}$ takes the form (Absil et al., 2008):

$$\mathbf{X}^{(t+1)} = R_{\mathbf{X}^{(t)}}(\alpha_t\xi_{\mathbf{X}^{(t)}}), \tag{2}$$

where $\xi_{\mathbf{X}^{(t)}} \in T_{\mathbf{X}^{(t)}}\mathcal{M}$ (namely, $\xi_{\mathbf{X}^{(t)}}$ is a tangent vector of $\mathcal{M}$ at $\mathbf{X}^{(t)}$) represents the search direction, α_t is the step size, and $R_{\mathbf{X}^{(t)}}(\cdot)$ represents the retraction at $\mathbf{X}^{(t)}$ which maps a tangent vector $\xi \in T_{\mathbf{X}^{(t)}}\mathcal{M}$ to a point on $\mathcal{M}$.

Tangent vectors serving as search directions are generally gradient-related. The gradient of a function $f(\mathbf{X})$ defined on $\mathcal{M}$, denoted as $\mathrm{Grad}f(\mathbf{X})$, depends on the Riemannian metric, which is a family of smoothly varying inner products on tangent spaces, i.e., $\langle\xi,\eta\rangle_\mathbf{X}$, where $\xi, \eta \in T_\mathbf{X}\mathcal{M}$ for any $\mathbf{X} \in \mathcal{M}$. The Riemannian gradient $\mathrm{Grad}f(\mathbf{X}) \in T_\mathbf{X}\mathcal{M}$ is the unique tangent vector that satisfies

$$\langle\mathrm{Grad}f(\mathbf{X}), \xi\rangle_\mathbf{X} = Df(\mathbf{X})[\xi], \tag{3}$$

for any $\xi \in T_\mathbf{X}\mathcal{M}$, where $Df(\mathbf{X})[\xi]$ represents the directional derivative of $f(\mathbf{X})$ in the tangent direction ξ.

2.2.1. EIGS via Riemannian Gradient

The constraint set in problem (1) constitutes a Stiefel manifold, i.e., $\mathrm{St}(n, q) = \{\mathbf{X} \in \mathbb{R}^{n\times q} : \mathbf{X}^\top\mathbf{X} = \mathbf{I}\}$, which turns the problem into a Riemannian one:

$$\max_{\mathbf{X}\in\mathrm{St}(n,q)} f(\mathbf{X}),$$

where $f(\mathbf{X}) \triangleq (1/2)\mathrm{tr}(\mathbf{X}^\top\mathbf{A}\mathbf{X})$. Under the canonical metric $\langle\xi,\eta\rangle_\mathbf{X} = \mathrm{tr}(\xi^\top(\mathbf{I} - \frac{1}{2}\mathbf{X}\mathbf{X}^\top)\eta)$ and by (3), the Riemannian gradient of $f(\mathbf{X})$ is

$$\mathrm{Grad}f(\mathbf{X}) = (\mathbf{I} - \mathbf{X}\mathbf{X}^\top)\mathbf{A}\mathbf{X}.$$

Furthermore, we use the Cayley transformation based retraction (Wen & Yin, 2013):

$$R_\mathbf{X}(\xi) = (\mathbf{I} - \frac{1}{2}\mathbf{S}(\xi))^{-1}(\mathbf{I} + \frac{1}{2}\mathbf{S}(\xi))\mathbf{X}, \tag{4}$$

for any $\xi \in T_\mathbf{X}\mathrm{St}(n, q)$, where $\mathbf{S}(\xi) = (P_\mathbf{X}\xi)\mathbf{X}^\top - \mathbf{X}(P_\mathbf{X}\xi)^\top$ and $P_\mathbf{X} = \mathbf{I} - \frac{1}{2}\mathbf{X}\mathbf{X}^\top$.

Given a line search method for determining the step size such as Amijo-Wolf conditions (Nocedal & Wright, 2006) or non-monotone line search (Wen & Yin, 2013), we can arrive at the **R**iemannian **G**radient **EIG**en**S**olver (RG-EIGS)

$$\mathbf{X}^{(t+1)} = R_{\mathbf{X}^{(t)}}(\alpha_t\mathrm{Grad}f(\mathbf{X}^{(t)})).$$

3. Doubly Stochastic Riemannian Gradient

In this section, we propose a doubly stochastic Riemannian gradient eigensolver, denoted as DSRG-EIGS, which generalizes Euclidean SGA and SCA to Stiefel manifolds and

[1] The given matrix $\mathbf{A}$ is assumed to be symmetric throughout the paper, i.e., $\mathbf{A}^T = \mathbf{A}$.

meanwhile extends RG-EIGS to the doubly stochastic optimization setting.

One update of the stochastic Riemannian gradient ascent takes the form (Bonnabel, 2013):

$$\mathbf{X}^{(t+1)} = R_{\mathbf{X}^{(t)}}(\alpha_t G(y_t, \mathbf{X}^{(t)})),$$

where $\alpha_t > 0$, y_t is an observation of the random variable y that follows some distribution and satisfies $\mathbb{E}[f(y, \mathbf{X})] = f(\mathbf{X})$, and $G(y, \mathbf{X}) \in T_{\mathbf{X}}\mathrm{St}(n, p)$ is a stochastic Riemannian gradient such that $\mathbb{E}[G(y, \mathbf{X})] = \mathrm{Grad}f(\mathbf{X})$.

3.1. Sampling over Data

We first consider the stochastic Riemannian gradient based on sampling over data. The given matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ can be written as a matrix summation. Although this summation could be made quite general, in our case, it's based on the following partitioning of $\mathbf{A}$ into a block matrix of size $n_r \times n_c$ for simplicity:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1n_c} \\ \cdots & \cdots & \cdots \\ \mathbf{A}_{n_r 1} & \cdots & \mathbf{A}_{n_r n_c} \end{pmatrix} = \sum_{k=1}^{n_r}\sum_{l=1}^{n_c} \mathbf{E}_{kl} \odot \mathbf{A},$$

where $\mathbf{E}_{kl} \in \{0,1\}^{n\times n}$ represents the element indicator of $\mathbf{A}_{kl}$ in $\mathbf{A}$. Define

$$f(s, \mathbf{X}) \triangleq p_s^{-1}\mathrm{tr}(\mathbf{X}^\top(\mathbf{E}_s \odot \mathbf{A})\mathbf{X})$$

and

$$G(s, \mathbf{X}) \triangleq p_s^{-1}(\mathbf{I} - \mathbf{X}\mathbf{X}^\top)(\mathbf{E}_s \odot \mathbf{A})\mathbf{X},$$

where s is a random variable taking on pair values from $\{(k, l) : k = 1, \cdots, n_r, l = 1, \cdots, n_c\}$, with respective probabilities $p_s > 0$ subject to $\sum_s p_s = 1$. It holds that $\mathbb{E}[f(s, \mathbf{X})] = f(\mathbf{X})$ and $\mathbb{E}[G(s, \mathbf{X})] = \mathrm{Grad}f(\mathbf{X})$.

We then get the stochastic Riemannian gradient $G(s, \mathbf{X})$ by sampling over data, which greatly reduces the complexity per iteration for data scanning, from that of a full scan, $O(n^2q)$ for dense matrices, to that of a partial scan. However, the complexity per iteration for updating variable $\mathbf{X}$ remains the same as that with the batch version RG-EIGS, i.e., $O(nq^2) + O(q^3)$. Hence, when q is large, it's still computationally cumbersome.

3.2. Sampling over Riemannian Gradient Coordinates

To further reduce the complexity per iteration, we now consider the stochastic Riemannian gradient based on sampling over Riemannian gradient coordinates $\{[\mathrm{Grad}f(\mathbf{X})]_{ij} : i = 1, \cdots, n, j = 1, \cdots, q\}$. This is exactly the idea of stochastic coordinate ascent (Nesterov, 2012). However, SCA is intended to solve unconstrained or separately constrained convex problems, and thus not suitable for ours, an inherently non-convex problem. In fact, the variable space $\mathrm{St}(n, q)$ (i.e., Stiefel manifold) and the gradient space $T_{\mathbf{X}}\mathrm{St}(n, q)$ (i.e., Euclidean space) are not the same one. Hence, the direct application of this method to our problem may be not well-defined, because a partial update of coordinates could make either $\mathbf{X}^{(t)}$ drift off the manifold, i.e., $\mathbf{X}^{(t)} \notin \mathrm{St}(n, q)$, or $\xi_{\mathbf{X}^{(t)}}$ step out of the tangent space, i.e., $\xi_{\mathbf{X}^{(t)}} \notin T_{\mathbf{X}^{(t)}}\mathrm{St}(n, q)$.

To tackle this issue, we propose to sample intrinsic coordinates of Riemannian gradients in the tangent space. Note that the tangent space of Stiefel manifold (Absil et al., 2008) at $\mathbf{X}$ can be explicitly represented as

$$T_{\mathbf{X}}\mathrm{St}(n, q) = \\ \{\mathbf{X}\mathbf{\Omega} + \mathbf{X}_\perp\mathbf{K} : \mathbf{\Omega}^\top = -\mathbf{\Omega} \in \mathbb{R}^{q\times q}, \mathbf{K} \in \mathbb{R}^{(n-q)\times q}\},$$

where $\mathbf{X}_\perp \in \mathbb{R}^{n\times(n-q)}$ represents the orthonormal complement of $\mathbf{X}$ in $\mathbb{R}^{n\times n}$ such that $(\mathbf{X}\ \mathbf{X}_\perp)$ is othogonal. By this representation, we can identify the intrinsic coordinates of a tangent vector $\xi_{\mathbf{X}}$ with corresponding $\mathbf{\Omega}$ and $\mathbf{K}$. We can also find the dimensionality of $\mathrm{St}(n, q)$ is $\frac{1}{2}q(q-1) + (n-q)q$.

Recall that our Riemannian gradient is $\mathrm{Grad}f(\mathbf{X}) = (\mathbf{I} - \mathbf{X}\mathbf{X}^\top)\mathbf{A}\mathbf{X}$, which can be rewritten as $\mathrm{Grad}f(\mathbf{X}) = \mathbf{X}_\perp\mathbf{X}_\perp^\top\mathbf{A}\mathbf{X}$. Hence, its intrinsic coordinates are $\mathbf{\Omega} = 0$ and $\mathbf{K} = \mathbf{X}_\perp^\top\mathbf{A}\mathbf{X}$. We only need to sample coordinates from $\mathbf{K}$. To gain advantages as with SCA, we sample columns of $\mathbf{K}$, which is equivalent to sample columns of $\mathbf{X}$. To this end, assume $\mathbf{X}$ is partitioned into a block matrix of size $1 \times q_c$ (i.e., column block matrix):

$$\mathbf{X} = (\mathbf{X}_{\cdot 1}, \mathbf{X}_{\cdot 2}, \cdots, \mathbf{X}_{\cdot q_c}) = \sum_{m=1}^{q_c} \mathbf{E}_{\cdot m} \odot \mathbf{X},$$

where $\mathbf{E}_{\cdot m} \in \{0,1\}^{n\times q}$ similarly represents the element indicator of $\mathbf{X}_{\cdot m}$ in $\mathbf{X}$. Define

$$f(r, \mathbf{X}) \triangleq p_r^{-1}\mathrm{tr}(\mathbf{X}^\top\mathbf{A}(\mathbf{E}_{\cdot r} \odot \mathbf{X}))$$

and

$$G(r, \mathbf{X}) \triangleq p_r^{-1}(\mathbf{I} - \mathbf{X}\mathbf{X}^\top)\mathbf{A}(\mathbf{E}_{\cdot r} \odot \mathbf{X}),$$

where r is a random variable taking on values $1, \cdots, q_c$, with respective probabilities $p_r > 0$ subject to $\sum_r p_r = 1$. It holds that $\mathbb{E}[f(r, \mathbf{X})] = f(\mathbf{X})$ and $\mathbb{E}[G(r, \mathbf{X})] = \mathrm{Grad}f(\mathbf{X})$. As we will see shortly, only one column block of $\mathbf{X}$ needs be updated at each step.

We now get the stochastic Riemannian gradient $G(r, \mathbf{X})$ by sampling over Riemannian intrinsic coordinates. It keeps $\mathbf{X}$ and $G(r, \mathbf{X})$ staying on the manifold and in the tangent space, respectively, and meanwhile the update step works like a Euclidean SCA step.

3.3. Doubly Stochastic Riemannian Gradient (DSRG)

By sampling over both data and intrisic Riemannian gradient coordinates, we arrive at our doubly stochastic Riemannian gradient $G(s,r,\mathbf{X}) \in T_{\mathbf{X}}\mathrm{St}(n,q)$:

$$G(s,r,\mathbf{X}) = p_s^{-1}p_r^{-1}(\mathbf{I} - \mathbf{X}\mathbf{X}^\top)(\mathbf{E}_s \odot \mathbf{A})(\mathbf{E}_{\cdot r} \odot \mathbf{X}).$$

It's easy to see that it is an unbiased estimate of the true Riemannian gradient, i.e., $\mathbb{E}[G(s,r,\mathbf{X})] = \mathrm{Grad} f(\mathbf{X})$. We then arrive at our DSRG ascent method:

$$\mathbf{X}^{(t+1)} = R_{\mathbf{X}^{(t)}}(\alpha_t G(s_t, r_t, \mathbf{X}^{(t)})). \qquad (5)$$

To simplify the above update, first let $g \triangleq G(s,r,\mathbf{X})$, $\tilde{\mathbf{U}} \triangleq (g, \mathbf{X})$ and $\tilde{\mathbf{V}} \triangleq (\mathbf{X}, -g)$. Since $g \in T_{\mathbf{X}}\mathrm{St}(n,q)$, we have $g^\top \mathbf{X} = 0$, which implies that $P_{\mathbf{X}}(g) = (\mathbf{I} - \frac{1}{2}\mathbf{X}\mathbf{X}^\top)g = g$ and thus $\mathbf{S}(g) = \tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top$. We then can write

$$\begin{aligned} R_{\mathbf{X}}(\alpha g) &= (\mathbf{I} - \frac{\alpha}{2}\mathbf{S}(g))^{-1}(\mathbf{I} + \frac{\alpha}{2}\mathbf{S}(g))\mathbf{X} \\ &= \mathbf{X} + \alpha\tilde{\mathbf{U}}(\mathbf{I} - \frac{\alpha}{2}\tilde{\mathbf{V}}^\top\tilde{\mathbf{U}})^{-1}\tilde{\mathbf{V}}^\top\mathbf{X}, \end{aligned}$$

by the Sherman-Morrison-Woodbury formula (Press et al., 2007). Note that

$$\tilde{\mathbf{V}}^\top\mathbf{X} = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{V}}^\top\tilde{\mathbf{U}} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -g^\top g & \mathbf{0} \end{pmatrix}.$$

Accordingly,

$$(\mathbf{I} - \frac{\alpha}{2}\tilde{\mathbf{V}}^\top\tilde{\mathbf{U}})^{-1} = \begin{pmatrix} \mathbf{W} & \frac{\alpha}{2}\mathbf{W} \\ -\frac{\alpha}{2}g^\top g\mathbf{W} & \mathbf{W} \end{pmatrix}$$

where $\mathbf{W} = (\mathbf{I} + \frac{\alpha^2}{4}g^\top g)^{-1}$. We then get

$$\begin{aligned} R_{\mathbf{X}}(\alpha g) &= \mathbf{X} + \alpha(\mathbf{I} - \frac{\alpha}{2}\mathbf{X}g^\top)g\mathbf{W} \qquad (6) \\ &= -\mathbf{X} + (\alpha g + 2\mathbf{X})\mathbf{W}. \end{aligned}$$

To see the properties of this method, let's focus on $\mathbf{W}$. Note that

$$g^\top g = \mathrm{diag}(\mathbf{0}, \cdots, \mathbf{0}, \mathbf{C}, \mathbf{0}, \cdots, \mathbf{0}),$$

where $\mathbf{C} = p_s^{-2}p_r^{-2}\mathbf{D}$ is the r^{th} diagonal block of $g^\top g$, and

$$\begin{aligned} \mathbf{D} &= \mathbf{X}_{\cdot r}^\top(\mathbf{E}_s \odot \mathbf{A})^\top(\mathbf{I} - \mathbf{X}\mathbf{X}^\top)(\mathbf{E}_s \odot \mathbf{A})\mathbf{X}_{\cdot r} \\ &= (\mathbf{A}_{kl}\mathbf{X}_{lr})^\top(\mathbf{A}_{kl}\mathbf{X}_{lr}) \\ &\quad - (\mathbf{X}_{k\cdot}^\top\mathbf{A}_{kl}\mathbf{X}_{lr})^\top(\mathbf{X}_{k\cdot}^\top\mathbf{A}_{kl}\mathbf{X}_{lr}), \end{aligned}$$

supposing $s = (k, l)$ (note that subscripts k, l, r are all block indices). Therefore, we get

$$\mathbf{W} = \mathrm{diag}(\mathbf{I}, \cdots, \mathbf{I}, \mathbf{B}^{-1}, \mathbf{I}, \cdots, \mathbf{I}),$$

where $\mathbf{B} = \mathbf{I} + \frac{\alpha^2}{4}\mathbf{C}$. We now can see that in (5) only the r^{th} column block of $\mathbf{X}$ needs be updated, while the left ones remain unchanged:

$$\mathbf{X}_{\cdot m} \leftarrow \begin{cases} (\alpha p_s^{-1}p_r^{-1}(\mathbf{H}^\top - \mathbf{X}\mathbf{X}_{k\cdot}^\top)\mathbf{A}_{kl}\mathbf{X}_{lm} \\ \qquad +2\mathbf{X}_{\cdot m})\mathbf{B}^{-1} - \mathbf{X}_{\cdot m}, & m = r \\ \mathbf{X}_{\cdot m}, & m \neq r \end{cases}$$

where $\mathbf{H} = (\mathbf{0}, \cdots, \mathbf{0}, \mathbf{I}, \mathbf{0}, \cdots, \mathbf{0})$ with $\mathbf{I}$ being the k^{th} column block.

Our DSRG-EIGS algorithm is summarized in Algorithm 1, which enjoys the advantages over RG-EIGS from the double stochasticity: 1) it achieves a greatly reduced complexity per iteration, $O(nq)$, especially when $\mathbf{A}$ is dense or q is large; 2) only a size-controlled small matrix $\mathbf{B}$ needs be inverted; 3) there is no need of matrix inversion when $q_c = q$, i.e., single column sampling over $\mathbf{X}$.

Algorithm 1 DSRG-EIGS

```
Input: $\mathbf{A}, T, \eta > 0, \zeta > 0$
Output: $\mathbf{X}^{(T)}$
Initialize $\mathbf{X}^{(0)}$ and $p_s = \frac{\|\mathbf{A}_s\|_F}{\sum_{\tilde{s}}\|\mathbf{A}_{\tilde{s}}\|_F}$ for any $s \in \mathcal{S} = \{(k,l) : k = 1, \cdots, n_r, l = 1, \cdots, n_c\}$.
for $t = 1, 2, \cdots, T$ do
    Sample $s_t = (k_t, l_t)$ from $\mathcal{S}$ according to $\{p_s\}$.
    Sample $r_t$ from $\{1, 2, \cdots, q_c\}$ uniformly.
    Set $\alpha_t = \frac{\eta}{1+\zeta t}$.
    Update $\mathbf{X}_{\cdot r_t}^{(t+1)} = -\mathbf{X}_{\cdot r_t}^{(t)} + (\alpha_t q_c p_{s_t}^{-1}(\mathbf{H}^{(t)^\top} - \mathbf{X}^{(t)}\mathbf{X}_{k_t\cdot}^{(t)^\top})\mathbf{A}_{k_t l_t}\mathbf{X}_{l_t r_t}^{(t)} + 2\mathbf{X}_{\cdot r_t}^{(t)})(\mathbf{B}^{(t)})^{-1}$
    and $\mathbf{X}_{\cdot r}^{(t+1)} = \mathbf{X}_{\cdot r}^{(t)}$ for $r \neq r_t$, where $\mathbf{B}^{(t)}$ and $\mathbf{H}^{(t)}$ are as defined in Section 3.3.
end for
```

4. Theoretical Analysis

We analyze convergence properties of Algorithm 1 in this section.

4.1. Local Convergence

We note the following facts. Stiefel manifold is smooth, connected and compact, with a positive global injectivity radius (Lee, 2012; Bonnabel, 2013). In addition, the function $f(\mathbf{X})$ to be maximized in our problem is three times continuously differentiable, the retraction (4) is twice continuously differentiable, and the stochastic Riemannian gradient (5) is unbiased, and bounded since both $\mathbf{A}$ and $\mathbf{X}$ are bounded. According to (Bonnabel, 2013), we have

Theorem 4.1. *If step sizes satisfy $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$, then for Algorithm 1, $f(\mathbf{X}^{(t)})$ converges almost surely and $\mathrm{Grad} f(\mathbf{X}^{(t)})$ converges to $\mathbf{0}$ almost surely, as $t \to \infty$.*

Note that only convergence to a local solution is guaranteed by Theorem 4.1.

4.2. Global Convergence

In fact, Theorem 4.1 can be strengthened to achieve a global convergence for our problem. Specifically, we investigate the squared cosine value of the principal angle between[2] $\mathbf{X}_t$ and the ground truth $\mathbf{V}$, which is defined as

$$\begin{aligned}\cos^2\langle\mathbf{X}_t,\mathbf{V}\rangle &\triangleq \lambda_{\min}(\mathbf{X}_t^\top\mathbf{V}\mathbf{V}^\top\mathbf{X}_t)\\ &= \min_{y\neq 0}\|\mathbf{V}^\top\mathbf{X}_t y\|_2^2/\|y\|_2^2.\end{aligned}$$

Note that if $\cos^2\langle\mathbf{X}_t,\mathbf{V}\rangle = 1$, then $\mathbf{X}_t = \mathbf{V}$ up to a $q\times q$ orthogonal matrix, that is, our goal is achieved. Actually we have the following strengthened theorem:

Theorem 4.2. *Define $\Theta_t = 1-\mathbb{E}[\cos^2\langle\mathbf{X}_t,\mathbf{V}\rangle]$. Assume that $\mathbf{A}$ has a positive eigen-gap, i.e., $\tau=\lambda_q-\lambda_{q+1}>0$, $\alpha_t=\frac{c}{t}$ with $c>\frac{2}{\tau}$, and $\cos^2\langle\mathbf{X}_s,\mathbf{V}\rangle\geq\frac{1}{2}$ with $s\geq 0$. Then we have $\Theta_t = O(\frac{1}{t})$ for $t\geq s$.*

Theorem 4.2 shows that our DSRG-EIGS algorithm converges to a global solution at a sub-linear rate in expectation. We note that the requirement on the initialization $\mathbf{X}_0$, which makes $\cos^2\langle\mathbf{X}_s,\mathbf{V}\rangle\geq\frac{1}{2}$ at certain iteration $s\geq 0$, is theoretically non-trivial. However, empirically a random initialization works well as we will observe in our experiments. Hence, Theorem 4.2 amounts to the convergence analysis at a later stage of the algorithm starting from $t_0 = s$ instead of $t_0 = 0$.

Similar to (Balsubramani et al., 2013), we have the following theorem which shows the concrete convergence rate of our algorithm. Before that, we define some stochastic quantities:

$$\mathbf{A}_t \triangleq p_{s_t}^{-1}(\mathbf{E}_{s_t}\odot\mathbf{A}),\quad \mathbf{Y}_t\triangleq p_{r_t}^{-1}(\mathbf{E}_{\cdot r_t}\odot\mathbf{X}_t),\quad\text{and}$$

$$\mathbf{Z}_t\triangleq\tilde{\mathbf{Z}}_t(\tilde{\mathbf{Z}}_t^\top\tilde{\mathbf{Z}}_t)^{-1/2},\ \text{where}\ \tilde{\mathbf{Z}}_t = \mathbf{X}_t+\alpha_t\mathbf{A}_t\mathbf{Y}_t.$$

Theorem 4.3. *Under the conditions of Theorem 4.2, assume that $a=c\tau$, $b=\gamma\mathbb{E}[\|\mathbf{A}_t\|_2^2]\mathbb{E}[\|\mathbf{Y}_t\|_2^2]$ with $\gamma>9$, and $t>s\geq 1$. Then it holds that*

$$\Theta_{t+1}\leq\Theta_s(\frac{s}{t+1})^a+\frac{4b}{a-1}(1+\frac{1}{s+1})^{a-1}\frac{1}{t+1}.$$

To prove Theorem 4.2-4.3, we need some lemmas.

Lemma 4.4. *Assume $\cos^2\langle\mathbf{X}_t,\mathbf{V}\rangle\geq\frac{1}{2}$ and let $\beta_t=\mathbb{E}[\|\mathbf{A}_t\|_2^2]\mathbb{E}[\|\mathbf{Y}_t\|_2^2]$. Then*

$$\begin{aligned}&1-\mathbb{E}[\cos^2\langle\mathbf{Z}_t,\mathbf{V}\rangle]\\&\leq(1-\alpha_t\tau)(1-\mathbb{E}[\cos^2\langle\mathbf{X}_t,\mathbf{V}\rangle])+5\beta_t\alpha_t^2+O(\alpha_t^3).\end{aligned}$$

Lemma 4.5.

$$\begin{aligned}&\cos^2\langle\mathbf{X}_{t+1},\mathbf{V}\rangle\\&\geq\cos^2\langle\mathbf{Z}_t,\mathbf{V}\rangle-4\alpha_t^2\|\mathbf{A}_t\|_2^2\|\mathbf{Y}_t\|_2^2\pm O(\alpha_t^3).\end{aligned}$$

Lemma 4.6. *Assume the constant $\gamma>9$. Then*

$$\Theta_{t+1}\leq(1-\alpha_t\tau)\Theta_t+\gamma\beta_t\alpha_t^2.$$

All the proofs are given in the supplementary. We notice that Theorem 4.3 and Lemma 4.6 provide a convenient form that enables us to leverage sampling distributions for improving the convergence rate.

4.3. Accelerated Global Convergence

Since the inequalities in Theorem 4.3 and Lemma 4.6 hold for general sampling distributions, we are able to improve the convergence rate by optimizing sampling distributions over data or gradient coordinates, i.e., importance sampling (Zhao & Zhang, 2014).

We only need to minimize β_t w.r.t. two sampling distributions, $\{p_s\}$ and $\{p_r\}$, which is equivalent to two independent problems:

$$\min_{\sum_s p_s=1}\mathbb{E}[\|\mathbf{A}_t\|_2^2]\ \text{ and }\ \min_{\sum_r p_r=1}\mathbb{E}[\|\mathbf{Y}_t\|_2^2].$$

Let $h(\{p_s\},\eta)$ be the Lagrange function for the first constrained optimization problem. Then

$$\begin{aligned}h(\{p_s\},\eta) &= \mathbb{E}[\|\mathbf{A}_t\|_2^2]+\eta(\sum_s p_s-1)\\ &= \sum_{k,l}p_{kl}^{-1}\|\mathbf{A}_{kl}\|_2^2+\eta(\sum_{k,l}p_{kl}-1),\\ \frac{\partial h}{\partial p_{kl}} &= -\frac{\|\mathbf{A}_{kl}\|_2^2}{p_{kl}^2}+\eta,\quad \frac{\partial^2 h}{\partial p_{kl}^2}=2\frac{\|\mathbf{A}_{kl}\|_2^2}{p_{kl}^3}>0.\end{aligned}$$

Setting $\frac{\partial h}{\partial p_{kl}}=0$, followed by normalization, yields the solution $p_{kl}^*=\|\mathbf{A}_{kl}\|_2/\sum_{k,l}\|\mathbf{A}_{kl}\|_2$. However, the spectral norm of a matrix is not quite easy to compute. We can relax[3] using an easy-to-compute upper bound of the spectral norm: $\min_{\sum_s p_s=1}\mathbb{E}[\|\mathbf{A}_t\|_2^2]\leq\min_{\sum_s p_s=1}\mathbb{E}[\|\mathbf{A}_t\|_F^2]$, and work on the latter problem. Likewise, we can find that $p_{kl}^*=\|\mathbf{A}_{kl}\|_F/\sum_{k,l}\|\mathbf{A}_{kl}\|_F$. This says that the blocks with a larger norm value should be sampled with a higher probability, which is quite a useful property for sparse matrix eigen-decomposition in that it can avoid frequent use of less informative blocks especially those zero or nearly zero blocks.

On the other hand, it holds that $p_r^*=\frac{\|\mathbf{X}_{\cdot r}\|_2}{\sum_r\|\mathbf{X}_{\cdot r}\|_2}$ similarly. Since $\|\mathbf{X}_{\cdot r}\|_2=1$ always, we get $p_r^*=q_c^{-1}$, which says that the optimal sampling over gradient coordinates turns out to be a uniform sampling. Two optimal sampling distributions are used in Algorithm 1.

5. Experimental Results

In this section, we empirically validate the effectiveness of our proposed doubly stochastic Riemannian gradient

[2] For notational convenience, we place iteration indices about t as subscripts hereafter.

[3] We acutally can use a tighter upper bound on spectral norm: $\|\mathbf{A}\|_2\leq\|\mathbf{A}\|_1\|\mathbf{A}\|_\infty$.

method for matrix eigen-decomposition, DSRG-EIGS, by comparing it with its deterministic counterpart, RG-EIGS (Wen & Yin, 2013), and using Matlab's EIGS function for benchmarking. Both DSRG-EIGS and RG-EIGS were implemented in Matlab on a machine with Windows OS, 8G of RAM.

5.1. Experimental Setting

We detail the experimental settings in this subsection, including the RG-EIGS implementation, initialization of $\mathbf{X}$ at $t = 0$, step size α_t, and quality measures for performance evaluation.

We adopt the original author's implementation for the RG-EIGS[4]. It uses the non-monotone line search with the well-known Barzilai-Borwein step size, which significantly reduces the iteration number, and performs well in practice. Both RG-EIGS and DSRG-EIGS are fed with the same initial value of $\mathbf{X}$, where each entry is sampled from the standard normal distribution $\mathcal{N}(0, 1)$ and then they all as a whole are orthogonalized. We set α_t for DSRG-EIGS to take the form of $\alpha_t = \frac{\eta}{1+\zeta t}$, where ζ is fixed to 2 throughout the experiments and η will be tuned.

The performance of different algorithms is evaluated using three quality measures: feasibilities $\|\mathbf{X}_t^\top \mathbf{X}_t - \mathbf{I}\|_F$, objective function values $\frac{1}{2}\mathrm{tr}(\mathbf{X}_t^\top \mathbf{A}\mathbf{X}_t)$ and squared cosine values of the principal angle between each iterate $\mathbf{X}_t$ and the ground truth $\mathbf{V}$, i.e., $\cos^2\langle \mathbf{X}_t, \mathbf{V}\rangle$. Lower values of feasibility are better, while large values of objective function and squared cosine are better. The output by EIGS is taken as the ground truth. We report the convergence curves of these measures, where the empirical convergence rate of each algorithm in terms of objective function values or squared cosine values can be observed.

5.2. Performance on Sparse Matrices

We first examine the performance of the algorithms on sparse matrices, which are downloaded from the university of Florida sparse matrix collection[5]. Their statistics are given in Table 1. Each of them is uniformly partitioned into a block matrix of size $m_r \times m_c$ given in Table 1. We use $q = 100$ and uniformly partition $\mathbf{X}$ into a block matrix of size $1 \times q_c$ with $q_c = q/2$.

Table 1. Sparse Matrices.

dataset	n	$\mathrm{nnz}(\mathbf{A})$	m_r	m_c
hangGlider	10,260	92,703	10	1
indef	64,810	565,996	50	1
IBMNA	169,422	1,279,274	150	1

Table 2. Dense Matrices.

dataset	n	$\mathrm{nnz}(\mathbf{A})$	m_r	m_c
citeseer	3,312	10,969,344	10	10
usps	9,298	86,452,804	20	20
pubmed	19,717	388,760,089	40	40
news20	19,928	397,125,184	40	40
a8a	32,561	1,060,218,721	40	40

The convergence curves of three quality measures for RG-EIGS and DSRG-EIGS on sparse matrices are shown in Figure 1, with one row of plots for each matrix and one column of plots for each measure. Each point on the convergence curve for RG-EIGS corresponds to one batch step[6], while it spans a fixed number of stochastic steps for DSRG-EIGS. We tested four different step sizes for DSRG-EIGS on each dataset. In each plot, the output of matlab's EIGS function, as a reference, is shown as a single point represented by a red pentagram. These performance results show that DSRG-EIGS consistently and significantly outperforms RG-EIGS in term of each quality measure. Specifically, DSRG-EIGS converges faster than RG-EIGS in terms of objective function values as shown in the middle column of plots, which clearly demonstrates the effectiveness and superiority of our algorithm to its deterministic version. Similar conclusions can be drawn for the right column of plots in terms of squared cosine values.

Moreover, we observe that the feasibility of RG-EIGS deteriorates in a manner similar to step functions. This is because that RG-EIGS relies heavily on the Sherman-Morrison-Woodbury formula which suffers from the numerical instability, and that the caused error will accumulate with iterations. In contrast, our DSRG-EIGS achieves a better feasibility especially on the first two sparse matrices, indicating that this issue is mitigated.

5.3. Performance on Dense Matrices

We now report the performance of the algorithms on dense matrices, which are RBF kernels generated using feature datasets: citeseer and pubmed[7], usps, news20 and a8a[8]. The statistics of resultant dense matrices are shown in Table 2, including their block sizes of uniform partitioning. We use $q = 10$ here. $\mathbf{X}$ is uniformly partitioned into $q_c = q/2$ column blocks as well.

Figure 2 shows the convergence curves of different measures on two dense matrices (results on left dense matrices are placed in the supplementary). As we can see,

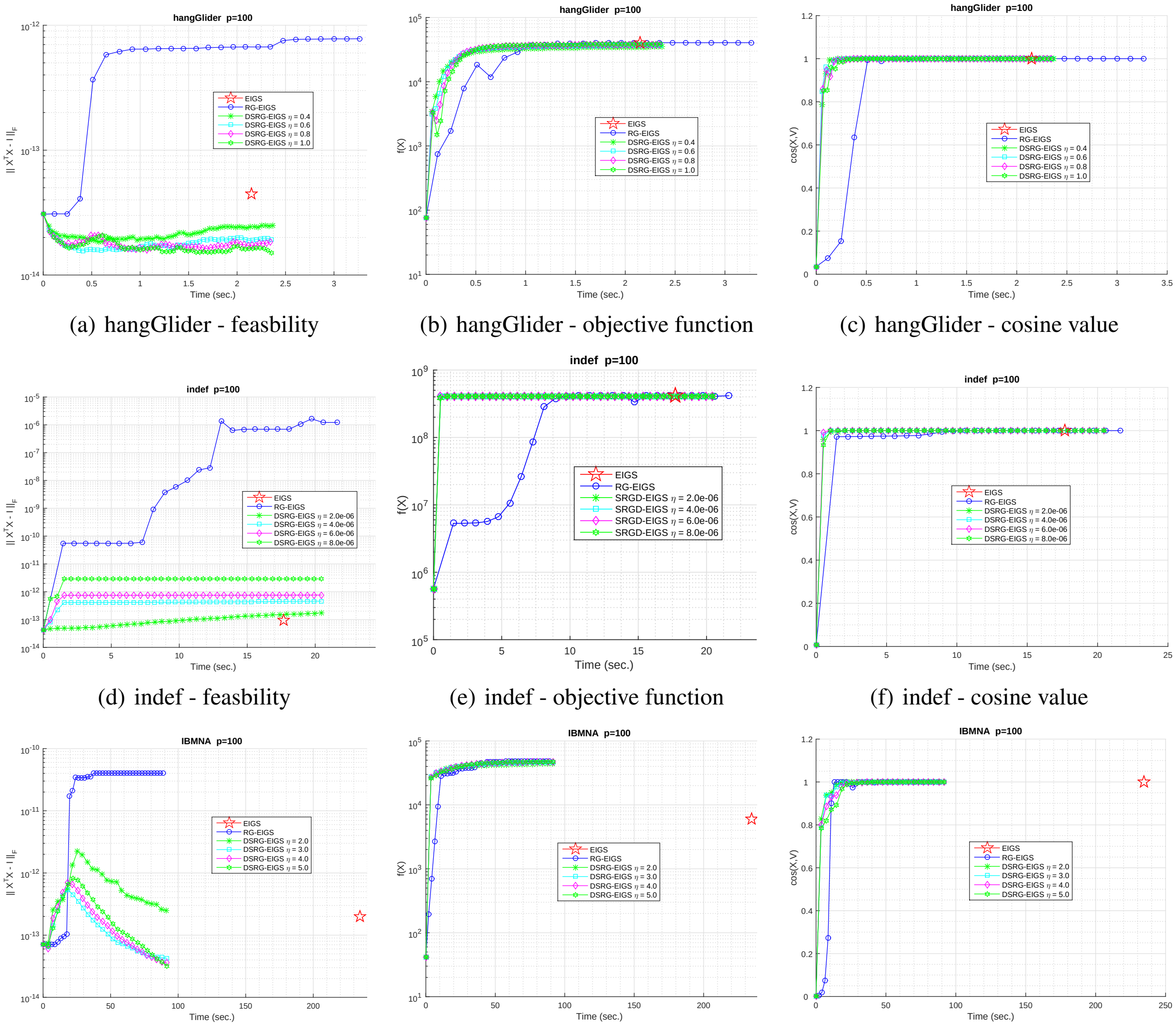
[4] optman.blogs.rice.edu/

[5] www.cise.ufl.edu/research/sparse/matrices/

[6] The decrease steps in Figure 1(b) are caused by the non-monotone step size used in the implementation of RG-EIGS.

[7] linqs.cs.umd.edu/projects/projects/lbc

[8] www.csie.ntu.edu.tw/~cjlin/libsvmtools



(a) hangGlider - feasbility (b) hangGlider - objective function (c) hangGlider - cosine value

(d) indef - feasbility (e) indef - objective function (f) indef - cosine value

(g) IBMNA - feasbility (h) IBMNA - objective function (i) IBMNA - cosine value

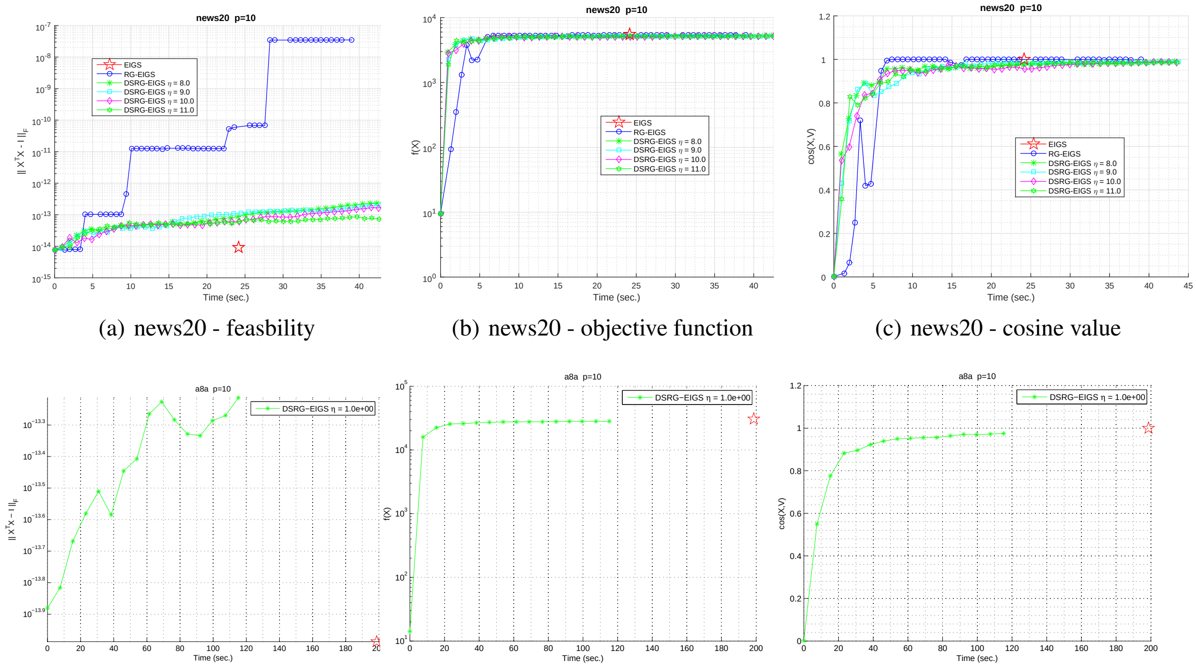
Figure 1. Performance on sparse matrices.

the performance of DSRG-EIGS is similar to the case of sparse matrices, compared to RG-EIGS. Especially on the a8a dataset, RG-EIGS fails to run due to running out of memory, while our DSRG-EIGS can still work (where the ground truth results were computed on a machine with the same CPU but a larger RAM). Therefore, the effectiveness and superiority of our algorithm are validated on dense matrices as well.

The experimental studies on both sparse and dense matrices demonstrate that the proposed algorithm is broadly effective and can be superior to its deterministic version. The advantages could be more pronounced in some cases. If the memory can not hold an input matrix, for example, a full matrix of size 32000 × 32000 like a8a, RG-EIGS clearly fails to run. In some real applications of matrix eigen-decomposition, when suboptimal solutions suffice to achieve satisfactory results in terms of third-party or domain-specific quality measures, such as modularity for spectral clustering, DSRG-EIGS would be a better choice than RG-EIGS.

6. Related Work

Typical existing approaches to matrix eigen-decomposition include the power method, the Lanczos algorithms, and Riemannian methods. The power method (Golub & Van Loan, 1996), finding the leading eigenpair (i.e., eigenvalue with the largest absolute value), starts from some initial vector, and then repeatedly alternates matrix-vector multiplication and vector normalization. Although it can be used on large sparse matrices, it may be slow and even diverge. Instead of disregarding the information in previous iterations as in power method, the Lanczos algorithm (Cullum & Willoughby, 2002) utilizes



(a) news20 - feasbility (b) news20 - objective function (c) news20 - cosine value

(d) a8a - feasbility (e) a8a - objective function (f) a8a - cosine value

Figure 2. Performance on dense matrices.

it to iteratively construct a basis of the Krylov subspace for eigen-decomposition. Riemannian methods address the problem from the Riemannian optimization perspective, such as optimization on Stiefel or Grassmann manifolds (Torbjorn Ringertz, 1997; Absil et al., 2008). One recently proposed method, Randomized SVD (Halko et al., 2011), finds the truncated SVD by random projections. All these methods perform the batch learning, while our focus in this paper is on stochastic algorithms. Another recent method, called MSEIGS (Si et al., 2014), tries to utilize graph cluster structure to speedup eigen-decomposition, while we consider more general matrices. The work most related to ours include online learning of eigenvectors (Garber et al., 2015), which only targets the leading eigenvector, i.e., $q = 1$, and coordinate descent on orthogonal matrices (Shalit & Chechik, 2014), which is a special case of Stiefel manifolds. (Garber et al., 2015) is based on the power method, and provides the regret analysis without empirical validation. We address the problem from a stochastic Riemannian optimization perspective. Stochastic coordinate descent is realized through Givens rotations with only local convergence guaranteed in (Shalit & Chechik, 2014), while we work on general Stiefel manifolds with global convergence guaranteed. On the other hand, doubly stochastic gradient has been used for scaling up kernel (Dai et al., 2014) and nonlinear component analysis (Xie et al., 2015), which rely on the primal feature data in vectors as with other PCA algorithms (Mitliagkas et al., 2013; Boutsidis et al., 2015), instead of relational data in square matrices as we target. In addition, importance sampling (Zhao & Zhang, 2014) has been considered for convex problems, while we extend its use on a non-convex problem in this paper.

7. Conclusion

We proposed the doubly stochastic Riemannian gradient ascent algorithm for matrix eigen-decomposition (DSRG-EIGS), i.e., a new eigensolver, which generalized the Euclidean stochastic gradient ascent and the Euclidean stochastic coordinate ascent to the Riemannian setting, or more precisely, Stiefel manifolds. The algorithm enjoys the advantages from both sides to achieve a greatly reduced complexity per iteration and be able to avoid the matrix inversion. We conducted a progressive convergence analysis, which shows that DSRG-EIGS converges to a global solution at a sub-linear rate in expectation, and that the convergence rate can be improved by leveraging sampling distributions. The effectiveness and superiority are verified on both sparse and dense matrices. For future work, we may address the limitations of DSRG-EIGS, including the non-trivial initialization and dependence on a positive eigen-gap. We may also conduct more empirical investigations on the algorithm.

Acknowledgments

We would like to thank Dr. Yiping Ke for reading the draft and offering useful suggestions, and anonymous reviewers for their valuable comments, which significantly improve the quality of the paper in different ways.

References

Absil, P-A, Mahony, Robert, and Sepulchre, Rodolphe. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

Balsubramani, Akshay, Dasgupta, Sanjoy, and Freund, Yoav. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems 26, December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pp. 3174–3182, 2013.

Bonnabel, Silvere. Stochastic gradient descent on riemannian manifolds. *IEEE Trans. Automat. Contr.*, 58(9): 2217–2229, 2013. doi: 10.1109/TAC.2013.2254619.

Boutsidis, Christos, Garber, Dan, Karnin, Zohar Shay, and Liberty, Edo. Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pp. 887–901, 2015.

Cullum, Jane K. and Willoughby, Ralph A. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. ISBN 0898715237.

Dai, Bo, Xie, Bo, He, Niao, Liang, Yingyu, Raj, Anant, Balcan, Maria-Florina, and Song, Le. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems 27, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3041–3049, 2014.

Drineas, Petros and Mahoney, Michael W. On the nyström method for approximating a gram matrix for improved kernel-based learning. *J. Mach. Learn. Res.*, 6:2153–2175, December 2005. ISSN 1532-4435.

Edelman, Alan, Arias, Tomás A., and Smith, Steven T. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, April 1999. ISSN 0895-4798. doi: 10.1137/S0895479895290954.

Garber, Dan, Hazan, Elad, and Ma, Tengyu. Online learning of eigenvectors. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 560–568, 2015.

Golub, Gene H. and Van Loan, Charles F. *Matrix Computations (3rd Ed.).* Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, May 2011. ISSN 0036-1445. doi: 10.1137/090771806.

Jolliffe, I. T. Principal component analysis. Hardcover, October 2002.

Lee, John M. *Introduction to smooth manifolds*. Springer, 2012.

Mitliagkas, Ioannis, Caramanis, Constantine, and Jain, Prateek. Memory limited, streaming pca. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2886–2894. Curran Associates, Inc., 2013.

Nesterov, Yurii. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi: 10.1137/100802001.

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In Dietterich, T.G., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems 14*, pp. 849–856. MIT Press, 2002.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

Parlett, Beresford N. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998. ISBN 0-89871-402-8.

Press, William H., Teukolsky, Saul A., Vetterling, William T., and Flannery, Brian P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688.

Shalit, Uri and Chechik, Gal. Coordinate-descent for learning orthogonal matrices through givens rotations. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 548–556, 2014.

Si, Si, Shin, Donghyuk, Dhillon, Inderjit S, and Parlett, Beresford N. Multi-scale spectral decomposition of massive graphs. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2798–2806. Curran Associates, Inc., 2014.

Torbjorn Ringertz, U. Eigenvalues in optimum structural design. *Institute for Mathematics and Its Applications*, 92:135, 1997.

Wen, Zaiwen and Yin, Wotao. A feasible method for optimization with orthogonality constraints. *Math. Program.*, 142(1-2):397–434, 2013. doi: 10.1007/s10107-012-0584-1.

Wen, Zaiwen, Yang, Chao, Liu, Xin, and Zhang, Yin. Trace-penalty minimization for large-scale eigenspace computation. Technical report, RICE UNIV HOUSTON TX DEPT OF COMPUTATIONAL AND APPLIED MATHEMATICS, 2013.

Xie, Bo, Liang, Yingyu, and Song, Le. Scale up nonlinear component analysis with doubly stochastic gradients. *CoRR*, abs/1504.03655, 2015.

Zhao, Peilin and Zhang, Tong. Stochastic optimization with importance sampling. *CoRR*, abs/1401.2753, 2014.