A Practical Riemannian Algorithm for Computing Dominant Generalized Eigenspace

Zhiqiang Xu, Ping Li
Cognitive Computing Lab, Baidu Research
No.10 Xibeiwang East Road, Beijing, 10085, China
10900 NE 8th St, Bellevue, WA 98004, USA
{xuzhiqiang04,liping11}@baidu.com

Abstract

Dominant generalized eigenspace computation, concerned with how to find one of the top-k generalized eigenspaces of a pair of real symmetric matrices, is one of the fundamental problems in scientific computing, data analysis, and statistics. In this work, we propose a practical Riemannian algorithm based on the first-order optimization on generalized Stiefel manifolds while efficiently leveraging second-order information. Particularly, we use inexact Riemannian gradients which result from running a fast least-squares solver to approximate matrix multiplications for avoiding costly matrix inversions involved therein. We also conduct a novel theoretical analysis, achieving a unified linear convergence rate regardless of the conventional generalized eigenvalue gap which is the key parameter to the currently dichotomized analysis: gap-dependent or gap-free. The resulting linear rate, albeit not optimal, remains valid in full generality. Despite the simplicity, empirically, our algorithm as a block generalized eigensolver remarkably outperforms existing solvers.

1 INTRODUCTION

Dominant generalized eigenspace computation is one of fundamental problems in scientific and engineering computing (Golub and Van Loan, 2013; Saad et al., 2010), data analysis (Shi and Malik, 2000; Karampatziakis and Mineiro, 2014; Guarracino et al., 2009), and statistics (Mardia et al., 1979; Diaz-Garcia, 2011). For example, it plays crucial roles in high-dimensional statistical problems such as canonical correlation analysis (CCA) (Hotelling, 1936; Xu and Li, 2019), Fisher discriminant analysis (Mika et al., 1999), and sufficient dimension reduction (Cook and Ni, 2005) which are in turn the backbones of many downstream tasks including regression (Kakade and Foster, 2007), clustering (Chaudhuri et al., 2009), and word embedding (Dhillon et al., 2011). The goal is to find one of the top-k generalized eigenspaces for a pair of real symmetric matrices of the same size with one being positive definite, where k is the number of components desired. Compared to the standard dominant eigenspace computation on a single real symmetric matrix (Saad et al., 2010), there is an additional cost from handling the inversion of the positive definite matrix. If it is handled naively, e.g., inverting at the outset, the cost could be as high as $O(n^3)$, where n is the size of matrices. Despite recent research towards scalable algorithms with low per-iteration costs (Ge et al., 2016; Allen-Zhu and Li, 2017; Li et al., 2017; Bhatia et al., 2018; Chen et al., 2019), either practical implementation was not taken into account or the performance is not satisfactory in practice.

In this work, we propose a practical Riemannian algorithm based on the first-order optimization on generalized Stiefel manifolds while efficiently leveraging second-order information. Given a pair of real symmetric matrices $(\mathbf{A}, \mathbf{B}) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$, where $\mathbf{B}$ is positive definite, the problem of *dominant k-dimensional generalized eigenspace computation* is to find a subspace of $\mathbb{R}^n$ in metric $\mathbf{B}$ spanned by $(\mathbf{A}, \mathbf{B})$'s generalized eigenvectors[1] corresponding to k largest generalized eigenvalues. Particularly, it admits a Riemannian formulation and the underlying Riemannian problem can be written as follows (Absil et al., 2008):

$$\max_{\mathbf{X}\in \mathrm{gSt}_{\mathbf{B}}(n,k)} f(\mathbf{X}) = \frac{1}{2}\mathrm{tr}(\mathbf{X}^\top \mathbf{A}\mathbf{X}), \tag{1}$$

where $\mathrm{gSt}_{\mathbf{B}}(n,k) = \{\mathbf{X} \in \mathbb{R}^{n\times k} : \mathbf{X}^\top \mathbf{B}\mathbf{X} = \mathbf{I}\}$ is the so-called generalized Stiefel manifold (Absil et al., 2008; Yger et al., 2012). One then can deploy the first-order optimization (e.g., Riemannian gradient method) on this

[1] See relevant definitions in the 2nd paragraph of Section 6.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

smooth manifold, where we need to address the computationally expensive matrix inversion $\mathbf{B}^{-1}$ encountered in computing Riemannian gradients. Instead of inverting $\mathbf{B}$ directly, we follow Ge et al. (2016) to approximate matrix multiplications of the form $\mathbf{B}^{-1}\mathbf{X}_t$ for $\mathbf{X}_t \in \mathbb{R}^{n\times k}$ varying with iterations, by running a fast least-squares solver (e.g., Nesterov's accelerated gradient descent) for a few iterations with a specially chosen warm start. Due to the warm-start, the initial error of the least-squares subproblem can be represented by the current negative Riemannian gradient, giving rise to a neat interpretation that the initial error gets increasingly close to zero during iterations. Meantime, the final error only needs to be commensurate with that of the current iterate and accordingly a few iterations are often sufficient for the least-squares solver in practice. In experiments, we use the preconditioned conjugate gradient method as the subproblem solver. Moreover, we leverage second-order information, by using the *inexact* Riemannian Barzilai-Borwein (RBB) step-sizes (Iannazzo and Porcelli, 2017), to bring ease of use and faster convergence simultaneously. The resulting algorithm, denoted as *rgGenElinK*, is almost as simple as the inexact power method (Ge et al., 2016), i.e., GenElinK, but remarkably outperforms the baseline algorithms including GenElinK, especially for $k > 1$. Furthermore, we conduct a different theoretical analysis from existing ones, achieving a unified linear rate of convergence regardless of the generalized eigenvalue gap, i.e., the key parameter to the currently dichotomized analysis: gap-dependent (convergence rate depends on the k-th generalized eigenvalue gap) or gap-free (the resulting rate does not depend on any gap). The resulting linear rate, albeit not optimal, remains valid in full generality similar to the gap-free rate. Our general and unified linear convergence analysis is built upon the structure of the solution space which can be characterized by two unique[2] top-k' and top-k'' generalized eigenspaces due to the positive generalized eigenvalue gaps at k' and k'', where $k' \leq k \leq k''$. The rgGenElinK algorithm is also applicable to the k-CCA problem. It is worth mentioning that the resulting algorithm, called *rgCCALin*, does not need to use a block size of double k for a top-k canonical subspace pair, and thus is more memory efficient.

The rest of the paper is organized as follows. Section 2 discusses literature work. Section 3 briefly introduces Riemannian geometry and optimization and Section 4 describes the k-CCA problem. The proposed algorithm is detailed in Section 5 and followed by analysis in Section 6. Section 7 presents experimental results and Section 8 concludes the paper.

[2]When $k' = k = k''$ they collapse into one unique top-k generalized eigenspace.

2 RELATED WORK

We discuss the main literature work on the dominant generalized eigenspace computation. First note that the scalable algorithms for standard eigenspace computation such as Xu et al. (2017); Xu and Gao (2018) are not applicable here due to the difference between two problems we mentioned in Section 1. The classic Lanczos algorithm can be used for matrix pairs as well, e.g., Algorithm 9.1 in Saad (2011), as long as matrix inversions can be handled similarly to ours. However, Lanczos algorithms require a considerably large amount of memory. Besides, theoretical support is lacking on the subproblem solver, warm start, and accuracy of the subproblem. Moreover, the default algorithm in MATLAB uses Cholesky decomposition of the positive definite matrix $\mathbf{B}$ such that it is converted into a standard eigenvalue problem. But it works only for medium-sized problems due to the Cholesky decomposition. In contrast, Ge et al. (2016) proposed the GenELinK algorithm with theoretical guarantees which is based on the inexact block power method. But the theoretical guarantee is unknown when the k-th generalized eigenvalue gap vanishes, and empirically, it performs not well enough as can be seen in our experiments. The doubly accelerated method (Allen-Zhu and Li, 2017) uses the shift-and-invert preconditioning paradigm, which aims at a top-1 generalized eigenvector, as the meta algorithm to recursively find a top-k generalized eigenspace via deflation, instead of outputting a top-k generalized eigenspace at a time. Despite the optimal rates of convergence, it skips practical implementations which, in fact, would be a concern due to many tuning parameters. Especially, the gap parameter is difficult to set in general. Moreover, the block solver via deflation may not always be applicable in some circumstances. Chen et al. (2018) considered the online computation of dominant generalized eigenspace and proposed a stochastic primal-dual algorithm with asymptotic guarantees on convergence rate and sample complexity. However, it assumes that the given matrices $\mathbf{A}, \mathbf{B}$ are commutative, which is unrealistic. Also, as we will see in our experiments, it works not well in practice. Bhatia et al. (2018) extended Oja's algorithm to the generalized streaming eigenvector computation ($k = 1$), which is quite a different setting from ours.

3 RIEMANNIAN GEOMETRY AND OPTIMIZATION

Let $\mathcal{M}$ be a Riemmanian manifold (Lee, 2012) of dimension d and $T_{\mathbf{X}}\mathcal{M}$ be its tangent space at $\mathbf{X} \in \mathcal{M}$ which is a d-dimensional Euclidean space $\mathbb{R}^d$ tangential to $\mathcal{M}$ at $\mathbf{X}$. $\mathcal{M}$ is often associated with a 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 of a function $f(\mathbf{X})$ on $\mathcal{M}$ is the unique tangent vector, i.e., $\tilde{\nabla} f(\mathbf{X}) \in T_{\mathbf{X}}\mathcal{M}$, that satisfies $\langle \tilde{\nabla} f(\mathbf{X}), \xi \rangle_{\mathbf{X}} = Df(\mathbf{X})[\xi]$ for any $\xi \in T_{\mathbf{X}}\mathcal{M}$, where $Df(\mathbf{X})[\xi]$ represents the directional derivative of $f(\mathbf{X})$ in ξ. The update of the Riemannian gradient ascent method (Absil et al., 2008) on $\mathcal{M}$ can be written as $\mathbf{X}_{t+1} = R(\mathbf{X}_t, \alpha_t \tilde{\nabla} f(\mathbf{X}_t))$, where $\alpha_t > 0$ is the step-size at the current step, and $R(\mathbf{X}_t, \cdot)$ represents the retraction at $\mathbf{X}_t$ that maps a tangent vector $\xi \in T_{\mathbf{X}_t}\mathcal{M}$ to a point on $\mathcal{M}$. Instead of using the costly exponential map, cheap retractions can be used. In addition, tangent vectors at different points need to be parallel transported to the same tangent space before arithmetic operations between them in theory. In practice, parallel transport is often omitted for computational concern.

For the generalized Stiefel manifold $\mathrm{gSt}_{\mathbf{B}}(n,k)$, the Riemannian metric can be chosen as $\langle \xi, \zeta \rangle_{\mathbf{B}} = \mathrm{tr}(\xi^\top \mathbf{B} \zeta)$, for any $\xi, \zeta \in T_{\mathbf{X}}\mathrm{gSt}_{\mathbf{B}}(n,k)$ and any $\mathbf{X} \in \mathrm{gSt}_{\mathbf{B}}(n,k)$. The objective function of Problem (1) under this metric has Riemannian gradient as $\tilde{\nabla} f(\mathbf{X}) = (\mathbf{B}^{-1} - \mathbf{X}\mathbf{X}^\top)\mathbf{A}\mathbf{X}$, by definition. We use the retraction defined by the generalized polar decomposition:

$$R(\mathbf{X}, \xi) = (\mathbf{X} + \xi)(\mathbf{I} + \xi^\top \mathbf{B} \xi)^{-1/2}$$

for $\xi \in T_{\mathbf{X}}\mathrm{gSt}_{\mathbf{B}}(n,k)$, which can be implemented by the modified Gram-Schmidt process with inner product $\langle \cdot, \cdot \rangle_{\mathbf{B}}$, denoted as $\mathrm{GS}_{\mathbf{B}}(\cdot)$.

4 CANONICAL CORRELATION ANALYSIS

Given a data pair $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \in \mathbb{R}^{d_x \times n} \times \mathbb{R}^{d_y \times n}$, let $\mathbf{C}_{xy} = \frac{1}{n}\widetilde{\mathbf{X}}\widetilde{\mathbf{Y}}^\top$, $\mathbf{C}_{xx} = \frac{1}{n}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top + r_x \mathbf{I}$, and $\mathbf{C}_{yy} = \frac{1}{n}\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^\top + r_y \mathbf{I}$, where $r_x, r_y > 0$ are regularization parameters for avoiding ill-conditioned matrices. The k-CCA problem is to solve the following program:

$$\max_{\mathbf{\Phi}^\top \mathbf{C}_{xx} \mathbf{\Phi} = \mathbf{\Psi}^\top \mathbf{C}_{xx} \mathbf{\Psi} = \mathbf{I}} \mathrm{tr}(\mathbf{\Phi}^\top \mathbf{C}_{xy} \mathbf{\Psi}).$$

The ground-truth solution to this problem, called the top-k canonical subspace pair, can be given by $(\mathbf{\Phi}^\star, \mathbf{\Psi}^\star) = (\mathbf{C}_{xx}^{-\frac{1}{2}}\widetilde{\mathbf{U}}, \mathbf{C}_{yy}^{-\frac{1}{2}}\widetilde{\mathbf{V}})$, where $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ are the top-k left and right singular subspaces of the whitened cross-covariance matrix $\mathbf{C}_{xx}^{-\frac{1}{2}}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-\frac{1}{2}}$, respectively. With[3]

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{C}_{xy} \\ \mathbf{C}_{xy}^\top & \mathbf{0} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix}, \ \mathbf{X} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{\Phi} \\ \mathbf{\Psi} \end{pmatrix},$$

the k-CCA problem is equivalent (Zhang, 2015) to Problem (1).

[3] We can manipulate the given data $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}})$ without the need to explicitly construct those matrices in our algorithm.

Algorithm 1 rgGenELinK

1: **Input:** a pair of real $n \times n$ symmetric matrices $(\mathbf{A}, \mathbf{B})$ where $\mathbf{B}$ is positive definite, k, initial step-size α_0, least-squares solver $\mathrm{ls}(\mathbf{A}, \mathbf{b}, \mathbf{x}_0)$
2: **Output:** the last iterate $\mathbf{X}_T$
3: $\mathbf{X}_0 \leftarrow \mathrm{GS}_{\mathbf{B}}(\mathbf{X}_0)$ where $\mathbf{X}_0 \in \mathbb{R}^{n \times k}$ is entry-wise i.i.d standard normal
4: **for** $t = 0, 1, 2, \cdots$ **do**
5: $\quad \widehat{\nabla f}_t \leftarrow \mathrm{ls}(\mathbf{B}, \mathbf{A}\mathbf{X}_t, \mathbf{X}_t^{(0)})$ which solves
$$\min_{\mathbf{X} \in \mathbb{R}^{n \times k}} l_t(\mathbf{X}) = \mathrm{tr}(\frac{1}{2}\mathbf{X}^\top \mathbf{B}\mathbf{X} - \mathbf{X}^\top \mathbf{A}\mathbf{X}_t)$$
$\quad$ with initial $\mathbf{X}_t^{(0)} = \mathbf{X}_t(\mathbf{X}_t^\top \mathbf{B}\mathbf{X}_t)^{-1}\mathbf{X}_t^\top \mathbf{A}\mathbf{X}_t$
6: $\quad \widehat{\tilde{\nabla} f}_t \leftarrow (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^\top \mathbf{B})\widehat{\nabla f}_t$
7: $\quad$ **if** t is even **then**
8: $\quad\quad \alpha_t \leftarrow \frac{\|\mathbf{X}_t - \mathbf{X}_{t-1}\|_F^2}{|\mathrm{tr}((\mathbf{X}_t - \mathbf{X}_{t-1})^\top (\widehat{\tilde{\nabla} f}_t - \widehat{\tilde{\nabla} f}_{t-1}))|}$ if $t > 0$
9: $\quad$ **else**
10: $\quad\quad \alpha_t \leftarrow \frac{|\mathrm{tr}((\mathbf{X}_t - \mathbf{X}_{t-1})^\top (\widehat{\tilde{\nabla} f}_t - \widehat{\tilde{\nabla} f}_{t-1}))|}{\|\mathbf{X}_t - \mathbf{X}_{t-1}\|_F^2}$
11: $\quad$ **end if**
12: $\quad \mathbf{X}_{t+1} \leftarrow \mathrm{GS}_{\mathbf{B}}(\mathbf{X}_t + \alpha_t \widehat{\tilde{\nabla} f}_t)$
13: $\quad$ terminate if stopping criterion is met
14: **end for**

5 PROPOSED ALGORITHM

The pseudo code of the proposed rgGenELinK algorithm is described in Algorithm 1. In Line 5, we only need to solve the sub-problem approximately by running a least-squares solver for a few iterations with the chosen warm-start $\mathbf{X}_t^{(0)}$. The warm-start plays an important role here, because it injects useful information about $\mathbf{X}_t$ from the previous iteration into the solver. After getting $\widehat{\nabla f}_t \approx \nabla f_t = \mathbf{B}^{-1}\mathbf{A}\mathbf{X}_t$ (∇f_t represents the exact solution to the sub-problem), Line 6 calculates the inexact Riemannian gradient according to $\tilde{\nabla} f(\mathbf{X}) = (\mathbf{I} - \mathbf{X}\mathbf{X}^\top \mathbf{B})\mathbf{B}^{-1}\mathbf{A}\mathbf{X}$. We then calculate the inexact Riemannian Barzilai-Borwein (RBB) step-size (Barzilai and Borwein, 1988; Iannazzo and Porcelli, 2017), except for the first iteration which needs to input an initial step-size α_0. The inexactness of the RBB step-sizes originates from three aspects: 1) a globalization strategy is ignored that ensures the global convergence (Raydan, 1997; Wen and Yin, 2013; Iannazzo and Porcelli, 2017); 2) parallel transport of old Riemannian gradients (Iannazzo and Porcelli, 2017) is omitted; 3) inexact Riemannian gradients are used. Despite multiple sources of inexactness, the RBB step-size scheme works well as is demonstrated in experiments. Empirically, $f(\mathbf{X}_t)$ does not necessarily keep increasing at each step due to the RBB step-size be-

ing non-monotone, but overall it will be increasing fast.

It is straightforward to apply the rgGenELinK to the k-CCA problem for top-k canonical subspaces. We can run the rgGenElinK for the matrix pair $(\mathbf{A}, \mathbf{B})$ defined in Section 4. Denote the output as $(\widehat{\mathbf{\Phi}}^\top, \widehat{\mathbf{\Psi}}^\top)^\top$. We then get the approximate top-k canonical subspace pair $(\widehat{\mathbf{\Phi}}(\widehat{\mathbf{\Phi}}^\top \mathbf{C}_{xx} \widehat{\mathbf{\Phi}})^{-\frac{1}{2}}, \widehat{\mathbf{\Psi}}(\widehat{\mathbf{\Psi}}^\top \mathbf{C}_{yy} \widehat{\mathbf{\Psi}})^{-\frac{1}{2}})$. We denote the resulting algorithm as *rgCCALin*. Compared to the CCALin (Ge et al., 2016) which, as a projection based method, requires a block size of $2k$ for a top-k canonical subspace pair, our rgCCALin algorithm does not need to set the block size to $2k$ since it is a gradient search based method. Thus, it is more memory efficient.

6 ANALYSIS

In this section, we present a different analysis from existing ones for Problem (1), based on the structure of the solution space together with appropriate potential functions as well as the methodology of the Riemannian optimization. The theorem and lemmas are presented, while their proofs are placed in the supplementary material.

Structure of Solution Space Let $(\lambda_i, \mathbf{u}_i)$ be the i-th generalized eigenpair of $(\mathbf{A}, \mathbf{B})$ in descending order of generalized eigenvalues, i.e., $\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{B}\mathbf{u}_i$ and $\mathbf{u}_i^\top \mathbf{B}\mathbf{u}_j = \delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$ otherwise 0. Let $\mathbf{U}_j = [\mathbf{u}_1, \cdots, \mathbf{u}_j]$. $\mathbf{U}_j$'s column space in non-Euclidean metric $\mathbf{B}$, denoted as $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_j)$, will be a top-$j$ generalized eigenspace (Golub and Van Loan, 2013). Denote the j-th generalized eigenvalue gap as $\Delta_j = \lambda_j - \lambda_{j+1}$. Letting $\lambda_0 = +\infty$ and $\lambda_{n+1} = -\infty$, we have $\Delta_0 = +\infty$ and $\Delta_n = +\infty$. Note that

$$f(\mathbf{U}_k) = \lambda_1 + \cdots + \lambda_{k-1} + \lambda_k = \max_{\mathbf{X} \in \mathrm{gSt}_{\mathbf{B}}(n,k)} f(\mathbf{X}).$$

If the k-th gap is positive, i.e., $\Delta_k > 0$, $(\mathbf{A}, \mathbf{B})$'s top-k generalized eigenspace $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_k)$ is unique. Otherwise, the k-th generalized eigenvalue is repeated and the target eigenspace is not unique any more. Generally, there always exist two integers k', k'' crossing k such that $k' \le k''$ and

$$\begin{cases} \lambda_{k'} > \lambda_{k'+1}, \\ \lambda_i = \lambda_k, \qquad i = k'+1, \cdots, k'', \\ \lambda_{k''} > \lambda_{k''+1}, \end{cases}$$

and then k', k'' fall into one and only one out of the following categories:

$$\begin{aligned} &a) \quad k' = k = k'', \\ &b) \quad 0 = k' < k < k'' < n, \\ &c) \quad 0 < k' < k < k'' = n, \\ &d) \quad 0 < k' < k < k'' < n, \\ &e) \quad 0 = k' < k < k'' = n. \end{aligned}$$

For all categories, we can write the solution space uniformly as follows: $\mathcal{U} = \{\mathbf{U} \in \mathrm{gSt}_{\mathbf{B}}(n,k) : \mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k'}) \subset \mathrm{col}_{\mathbf{B}}(\mathbf{U}) \subset \mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k''})\}$. It is easy to see that $f(\mathbf{U}_k) = f(\mathbf{U})$ for any $\mathbf{U} \in \mathcal{U}$. The target eigenspace is unique only when the given matrix pair falls into Category a), otherwise it is not unique but both $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k'})$ and $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k''})$ are unique. Note that Category e) is trivial as $f(\mathbf{X})$ becomes a constant. We now can define the structured gap of generalized eigenvalues around k as $\Delta_\dagger = \min\{\Delta_{k'}, \Delta_{k''}\}$, which always remains positive. Note that this gap is different from the conventional generalized eigenvalue gap Δ_k. Our theoretical results will be stated in terms of $\Delta_\dagger$.

Potential Functions Motivated by the structure of the solution space $\mathcal{U}$, the progress of iterate $\mathbf{X}_t$ to $\mathcal{U}$ can be determined by its progress to the two unique subspaces $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k'})$ and $\mathrm{col}_{\mathbf{B}}(\mathbf{U}_{k''})$, where distances between subspaces of different dimensions need to be handled. For global convergence (Pitaval et al., 2015), we use the Martin distance (Ye and Lim, 2016) in metric $\mathbf{B}$. For any matrix $\mathbf{G} \in \mathbb{R}^{l \times s}$, we define for brevity that

$$\mathrm{Det}(\mathbf{G}) = \begin{cases} \det((\mathbf{G}\mathbf{G}^\top)^{1/2}), & l \le s \\ \det((\mathbf{G}^\top \mathbf{G})^{1/2}), & \text{else} \end{cases}.$$

Our potential function then is given by

$$\psi(\mathbf{X}, \mathcal{U}) = \max_{l = k' \text{ or } k''} -2 \log \mathrm{Det}(\mathbf{X}^\top \mathbf{B} \mathbf{U}_l).$$

We point out that only one distance in the potential function is active for Categories a)-c) and e). To be well-defined for $\log \mathrm{Det}(\mathbf{X}^\top \mathbf{B}\mathbf{U}_j)$ at $j = 0$, let $\log \mathrm{Det}(\mathbf{X}^\top \mathbf{B}\mathbf{U}_0) \triangleq 0$. If $\psi(\mathbf{X}, \mathcal{U}) = 0$, it must hold that $\mathbf{X} \in \mathcal{U}$. We now can state our theorem as follows.

Theorem 6.1 *If we use Nesterov's accelerated gradient descent method as the least-squares solver, then Algorithm 1 under a constant or mixed step-size scheme is able to converge globally to one of the ground-truth solutions to Problem (1), i.e., $\psi(\mathbf{X}_T, \mathcal{U}) < \epsilon$, with overall complexity*

$$O((\mathrm{nnz}(\mathbf{A}) + \mathrm{nnz}(\mathbf{B}) \sqrt{\kappa(\mathbf{B})} \log \frac{\lambda_1}{\Delta_\dagger})(\frac{\lambda_1}{\Delta_\dagger})^2 \log \frac{1}{\epsilon}),$$

where $\mathrm{nnz}(\mathbf{A})$ represents the number of nonzero entries in $\mathbf{A}$ and $\kappa(\mathbf{B})$ represents the condition number of $\mathbf{B}$.

We briefly describe the proof idea here and have the details deferred to the supplementary material. For proof, we show that the potential function eventually decays to zero at an exponential rate, which can be done by demonstrating that the potential will be contracted by a constant in $(0, 1)$ at each iteration. To this end, the potential function can be split into two parts: one part is about exact

updates and the other is about errors. The first part is handled by the following lemma.

Lemma 6.2 *Let* $\widehat{\mathbf{X}}_{t+1} = \mathbf{X}_t + \alpha_t \tilde{\nabla} f(\mathbf{X}_t)$. *If* $0 < \alpha_t < \frac{1}{8\lambda_1 \eta_{jt}}$ *where* $\eta_{jt} = \frac{\text{dist}_b^2(\mathbf{X}_t, \mathbf{U}_j)}{1 - \text{dist}_b^2(\mathbf{X}_t, \mathbf{U}_j)}$ *and distance functions* dist *are given in Lemma 6.4, then*

$$-2\log\text{Det}(\widehat{\mathbf{X}}_{t+1}^\top \mathbf{B}\mathbf{U}_j) \leq \text{dist}_m^2(\mathbf{X}_t, \mathbf{U}_j) - 2\alpha_t \text{dist}_f(\mathbf{X}_t, \mathbf{U}_j) + 32k\alpha_t^2\lambda_1^2\eta_{jt}^2.$$

The error part can be controlled with Lemma 6.3.

Lemma 6.3 *For the sub-problem in Algorithm 1, letting*

$$\xi_t(\mathbf{X}) = \mathbf{X} - \mathbf{X}_t^\star \text{ and } \epsilon_t(\mathbf{X}) = l_t(\mathbf{X}) - l_t(\mathbf{X}_t^\star),$$

where $\mathbf{X}_t^\star = \mathbf{B}^{-1}\mathbf{A}\mathbf{X}_t = \arg\min l_t(\mathbf{X})$, *we have that*

$$\epsilon_t(\mathbf{X}) = \frac{1}{2}\|\xi_t(\mathbf{X})\|_{\mathbf{B},F}^2 \text{ and } \epsilon_t(\mathbf{X}_t^{(0)}) = \frac{1}{2}\|\tilde{\nabla} f(\mathbf{X}_t)\|_{\mathbf{B},F}^2,$$

where the norm is defined in Lemma 6.6. In addition, Nesterov's accelerated gradient descent takes $O(\text{nnz}(\mathbf{A}) + \text{nnz}(\mathbf{B}))\sqrt{\kappa(\mathbf{B})}\log\frac{\epsilon_t(\mathbf{X}_t^{(0)})}{\epsilon_t(\widehat{\nabla f}_t)})$ *complexity to reach sub-optimality* $\epsilon_t(\widehat{\nabla f}_t)$.

In fact, we have $\xi_t(\mathbf{X}_t^0) = -\tilde{\nabla} f(\mathbf{X}_t)$, which gives rise to a neat interpretation of the initial error with the sub-problem and strong support for the chosen warm-start noting that the gradient gradually approaches zero. Particularly, the final error of the subproblem only needs to be as accurate as follows:

$$\epsilon_t(\widehat{\nabla f}_t) \propto \Delta_\dagger^2(1 - \min_{l=k' \text{ or } k''} \text{Det}^2(\mathbf{X}_t^\top \mathbf{B}\mathbf{U}_l)).$$

There are some other issues in proof that also need to be handled with the following lemmas.

Lemma 6.4 *For* $\mathbf{X} \in \text{gSt}_\mathbf{B}(n,k)$ *and* $\mathbf{Y} \in \text{gSt}_\mathbf{B}(n,l)$, *let*

$$\text{dist}_m(\mathbf{X},\mathbf{Y}) = (-2\log\text{Det}(\mathbf{X}^\top\mathbf{B}\mathbf{Y}))^{\frac{1}{2}},$$
$$\text{dist}_b(\mathbf{X},\mathbf{Y}) = (1 - \text{Det}^2(\mathbf{X}^\top\mathbf{B}\mathbf{Y}))^{\frac{1}{2}},$$
$$\text{dist}_c(\mathbf{X},\mathbf{Y}) = (\min\{k,l\} - \|\mathbf{X}^\top\mathbf{B}\mathbf{Y}\|_F^2)^{\frac{1}{2}}.$$

We then have that

$$\text{dist}_b(\mathbf{X},\mathbf{Y}) \leq \text{dist}_m(\mathbf{X},\mathbf{Y}),$$
$$\text{dist}_b(\mathbf{X},\mathbf{Y}) \leq \text{dist}_c(\mathbf{X},\mathbf{Y}) \leq (\min\{k,l\})^{\frac{1}{2}}\text{dist}_b(\mathbf{X},\mathbf{Y}).$$

Three distance functions are extended from the Martin, Binet-Cauchy, and Chordal distances (Ye and Lim, 2016), respectively, in order to accommodate distances between subspaces of different dimensions.

Lemma 6.5 *Let*

$$\text{dist}_f(\mathbf{X}_t, \mathbf{U}_j) = \begin{cases} f(\mathbf{U}_j) - f(\mathbf{X}_t\mathbf{P}_j), & j \leq k \\ f(\mathbf{U}_j\mathbf{Q}_j) - f(\mathbf{X}_t), & j \geq k \end{cases},$$

where $\mathbf{P}_{jt}$ *and* $\mathbf{Q}_{jt}$ *are from the rank-*$\min\{j,k\}$ *SVD:* $\mathbf{X}_t^\top\mathbf{B}\mathbf{U}_j = \mathbf{P}_{jt}\mathbf{\Sigma}_{jt}\mathbf{Q}_{jt}^\top$. *We then have that*

$$\text{dist}_f(\mathbf{X}_t, \mathbf{U}_j) \geq \Delta_j \text{dist}_b^2(\mathbf{X}_t, \mathbf{U}_j).$$

The above lemma is one key to the linear convergence especially for Categories b)-e) by relating to $\mathbf{U}_{k'}$ and $\mathbf{U}_{k''}$, because there is no non-trivial lower bound on $\text{dist}_f(\mathbf{X}_t, \mathbf{U})$ for any $\mathbf{U} \in \mathcal{U}$.

Lemma 6.6 *We have that* $\|\tilde{\nabla} f(\mathbf{X})\|_{\mathbf{B},2} \leq \lambda_1$ *and*

$$\|\tilde{\nabla} f(\mathbf{X})\|_{\mathbf{B},F}^2 \leq k\lambda_1^2 \min\{2\min_{\mathbf{U}\in\mathcal{U}} \text{dist}_m^2(\mathbf{X},\mathbf{U}), 1\},$$

where $\|\mathbf{X}\|_{\mathbf{B},2} = \|\mathbf{B}^{\frac{1}{2}}\mathbf{X}\|_2$ *and* $\|\mathbf{X}\|_{\mathbf{B},F}^2 = \text{tr}\left(\mathbf{X}^\top\mathbf{B}\mathbf{X}\right)$.

This lemma is yet another key to achieving a linear rate.

Lemma 6.7 *If* $a > 1$ *and* $a_t \leq (1 - \frac{a}{\nu+t})a_{t-1} + \frac{b}{(\nu+t)^2}$ *holds for* $t \geq 1$, *then (Balsubramani et al., 2013)*

$$a_t \leq (\frac{\nu+1}{\nu+1+t})^a a_0 + \frac{2^{a+1}b}{a-1}\frac{1}{\nu+1+t}.$$

Lemma 6.8 *For* $\mathbf{X} \in \text{gSt}_\mathbf{B}(n,k)$, $\mathbf{Y} \in \text{gSt}_\mathbf{B}(n,l)$, *let*

$$\Omega_+(\mathbf{Y}) = \{\mathbf{Z} \in \text{gSt}_\mathbf{B}(n,k) : \text{col}_\mathbf{B}(\mathbf{Y}) \subset \text{col}_\mathbf{B}(\mathbf{Z})\},\ l \leq k,$$
$$\Omega_-(\mathbf{Y}) = \{\mathbf{Z} \in \text{gSt}_\mathbf{B}(n,k) : \text{col}_\mathbf{B}(\mathbf{Z}) \subset \text{col}_\mathbf{B}(\mathbf{Y})\},\ l \geq k.$$

We then have that

$$\text{dist}_m(\mathbf{X},\mathbf{Y}) = \begin{cases} \min_{\mathbf{Z}\in\Omega_+(\mathbf{Y})} \text{dist}_m(\mathbf{X},\mathbf{Z}), & l \leq k \\ \min_{\mathbf{Z}\in\Omega_-(\mathbf{Y})} \text{dist}_m(\mathbf{X},\mathbf{Z}), & l > k \end{cases}.$$

Two sets $\Omega_+(\mathbf{Y})$ and $\Omega_-(\mathbf{Y})$ above are called Schubert varieties of subspace $\text{col}_\mathbf{B}(\mathbf{Y})$. This lemma offers us a natural and intrinsic way to analyze the distance between subspaces of different dimensions. It is extended from Ye and Lim (2016) by replacing the Euclidean metric $\mathbf{I}$ with non-Euclidean metric $\mathbf{B}$ for $\mathbb{R}^n$. It is used together with Lemma 6.6 to help get the linear rate.

Lemma 6.9 *If* $\mathbf{X}_0 = \mathbf{W}(\mathbf{W}^\top\mathbf{B}\mathbf{W})^{-1/2}$ *with* $\mathbf{W}$ *being entry-wise standard normal, then*

$$\text{dist}_m^2(\mathbf{X}_0, \mathbf{U}_j) < -2k\log\frac{\eta\sqrt{\kappa(\mathbf{B})}}{k + \sqrt{nk}}$$

with probability at least $1 - \eta$ *for any* $\eta \geq 0$.

Therefore, $\text{dist}_m^2(\mathbf{X}_0, \mathbf{U}_j) < +\infty$ with probability 1, which means $\mathbf{X}_0$ is neither a saddle point nor a minimizer almost surely. This explains the global convergence of Algorithm 1 mentioned in Theorem 6.1.

Before closing this section, a few remarks on Theorem 6.1 are in order. **1)** If the matrix pair falls into Categories a)-c) and e) then constant step-sizes are sufficient, otherwise mixed step-sizes are required (details can be found in the supplementary material). **2)** With the new gap $\Delta_\dagger$, we get a unified convergence rate for both the gap-dependent and gap-free cases that are defined with the conventional gap Δ_k. The dependence of our rate on $\Delta_\dagger$ extends the common dependence on Δ_k. Compared to other methods, e.g., linear dependence on Δ_k of the GenELink (Ge et al., 2016), it is quadratic and worse when $\Delta_\dagger = \Delta_k$. The quadratic dependence seems a phenomenon associated with offline gradient methods (Shamir, 2015, 2016). Nonetheless, our rate covers the vanishing gap $\Delta_k = 0$, while the rate in Ge et al. (2016) does not. Also, our rate remains valid across categories, similar to the gap-free rate (Allen-Zhu and Li, 2017) but from a different perspective.

7 EXPERIMENTS

We now examine our algorithms' empirical performance on synthetic and real data. All algorithms were implemented in MATLAB and fed with the same random initials for each dataset.

7.1 Generalized Eigenspace

We compare Algorithm 1 to the GenElinK (Ge et al., 2016) and the stochastic generalized Hebbian algorithm (SGHA) (Chen et al., 2018) for the dominant generalized eigenspace computation. The initial step-size for the rgGenElinK is set to $\alpha_0 = 10^{-3}$ unless otherwise stated. SGHA is a stochastic primal-dual algorithm which iterates the primal step and the dual step. The constraint of Problem (1) is handled by the dual update. One issue with the SGHA is that the step-size η needs to be set relatively very small, otherwise it will cause the floating-point overflow easily during iterations. To implement the SGHA, we evenly partition $\mathbf{A}$ and $\mathbf{B}$ into column blocks, and sample blocks uniformly at random to form $\mathbf{A}_t$ and $\mathbf{B}_t$ at each iteration. Both rgGenElinK and GenElinK have the same subproblems which we solved by the preconditioned conjugate gradient method with a matlab built-in function. For this sub-problem solver, four and ten iterations are used on synthetic and real data, respectively. For evaluation of the generalized eigenspace computation, the following quality measures are used:

- relative objective error

$$\text{rdist}_f(\mathbf{X}, \mathbf{U}_k) = \frac{f(\mathbf{U}_k) - f(\mathbf{X})}{f(\mathbf{U}_k)},$$

- squared sine value of the largest principal angle between $\mathbf{X}$ and $\mathbf{U}_k$, i.e.,

$$\sin^2 \theta_{\max}(\mathbf{X}, \mathbf{U}_k) = 1 - \sigma_{\min}^2(\mathbf{X}^\top \mathbf{B} \mathbf{U}_k),$$

For the above two measures (smaller is better), the ground-truth is obtained by MATLAB's eigs function.

7.2 Synthetic Data

Three synthetic datasets with $n = 1000, k = 4$ and generalized eigenvalue gap $\Delta \in \{0.16, 0.016, 0.0016\}$ are generated according to the simultaneous diagonalization equations[4]:

$$\mathbf{Z}^\top \mathbf{A} \mathbf{Z} = \mathbf{\Lambda} \quad \text{and} \quad \mathbf{Z}^\top \mathbf{B} \mathbf{Z} = \mathbf{I}.$$

We set

$$\mathbf{\Lambda} = \text{diag}(1 - \Delta[-0.4, -0.2, 0, 1, 0.1, 0.2, 0.3, 0.4], \frac{|a_1|}{n}, \cdots, \frac{|a_{n-8}|}{n})$$

with a_i being i.i.d. standard normal samples, and

$$\mathbf{Z} = \mathbf{Q}\text{diag}(5b_1 + 5, \cdots, 5b_n + 5)\mathbf{Q}^\top,$$

where b_i are i.i.d. uniform samples on $(0, 1)$ and $\mathbf{Q}$ is orthogonal. We then have that

$$\mathbf{A} = \mathbf{Z}^{-\top} \mathbf{\Lambda} \mathbf{Z}^{-1} \quad \text{and} \quad \mathbf{B} = \mathbf{Z}^{-\top} \mathbf{Z}^{-1}.$$

The SGHA uses block size 100 and step-sizes η are shown in legends. Figure 1 reports the performance of the algorithms on these datasets with varying generalized eigenvalue gaps. First, the results indicate that our rgGenElinK consistently outperforms all the others in terms of both measures. Second, for both rgGenElinK and GenElinK, larger gap Δ results in faster convergence. Third, the SGHA is not working well in all cases. Particularly, in the case of a small gap, ours is the only working algorithm, while the GenElinK can be even worse than the SGHA in terms of the second measure.

We also test on a pair of sparse symmetric random matrices with one being positive definite, generated by MATLAB's sprandsym function. The generating parameters, i.e., size $n = 10000$, density 10^{-3}, and reciprocal condition number 0.1, are used for $(\mathbf{A}, \mathbf{B})$ with $\mathbf{B}$ generated from the first kind of positive definiteness. We use $k = 20$ here and set the block size to 2000 for the SGHA. Algorithms' convergence behaviors on this dataset can be seen in Figure 2. Similar to the case of the small-gap datasets above, our rgGenElinK performs well, while others even can't converge in the given time interval.

[4] http://fourier.eng.hmc.edu/e161/lectures/algebra/node7.html

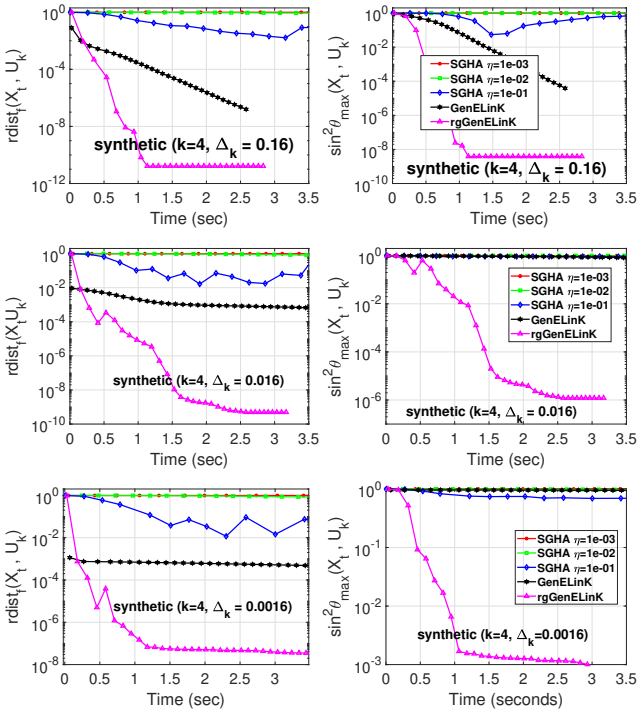

Figure 1: Performance of generalized eigensolvers on synthetic data - part I.

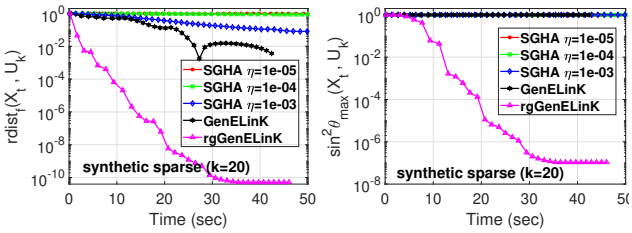

Figure 2: Performance of generalized eigensolvers on synthetic data - part II.

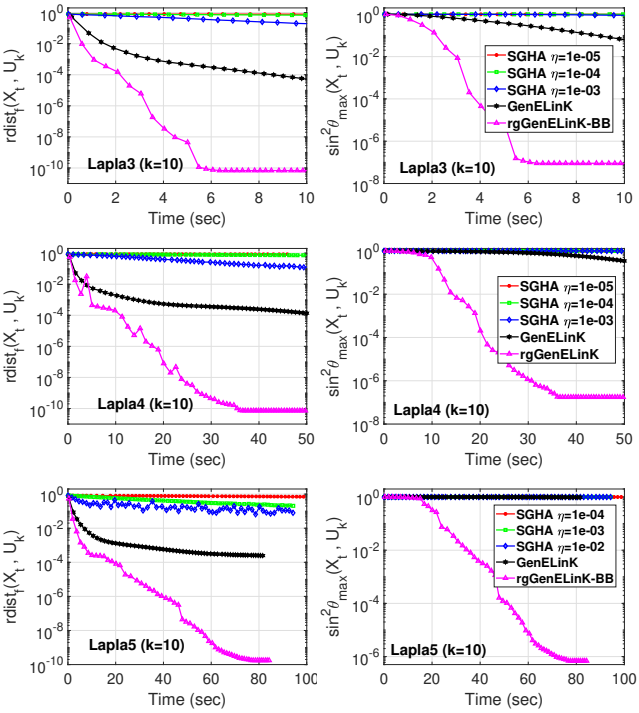

Figure 3: Performance of generalized eigensolvers on real data.

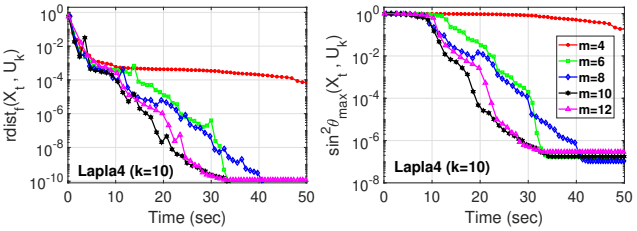

Figure 4: Performance of the rgGenELinK with $\alpha_0 = 10^{-3}$ and varying m.

7.3 Real Data

We turn to some challenging real data[5]. Challenges spring from the clustered eigenvalues which often result in small relative generalized eigenvalue gaps. Statistics of the data are given in Table 1. $k = 10$ is used and block sizes for the SGHA on the three datasets are $500, 2000, 4000$, respectively. As shown in Figure 3, our rgGenElinK is a clear winner over others and advantages are more pronounced in terms of the second measure.

Last, we examine the sensitivity of rgGenElinK's two parameters on dataset "Lapla4": initial step-size α_0 and iteration number m of the least-squares solver. Figure 4 shows the performance of the algorithm with fixed $\alpha_0 = 10^{-3}$ and varying m. We can see that if m is set to be too small, e.g., $m = 4$, poor results are observed probably due to the challenges with the data, though it is often sufficient for the synthetic data. We thus increased m in previous experiments to secure more accurate Riemannian gradients for faster convergence. In practice, we might also dynamically increase m starting from a small number, until acceptable progress per iteration is observed.

Table 1: Summary of real data

Matrix pair	n	nnz($\mathbf{A}$)	nnz($\mathbf{B}$)
Lapla3	5795	136565	141779
Lapla4	10891	259425	269639
Lapla5	18903	455337	489875

[5] http://faculty.smu.edu/yzhou/data/matrices.htm

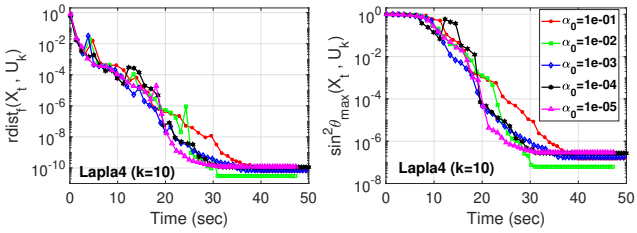

Figure 5: Performance of the rgGenELinK with $m = 10$ and varying α_0.

On the other hand, we fix $m = 10$ and test with varying α_0. As is shown in Figure 5, Algorithm 1 is fairly insensitive to the initial step-size. In addition to the automatic BB step-size, this further eases practitioners' concern about how to set the initial step-size, verifying the practical use of the proposed algorithm.

7.4 Canonical Subspaces

We compare our rgCCALin to the CCALin (Ge et al., 2016) and the S-AppGrad (Ma et al., 2015) for k-CCA. Note that the CCALin is an instance of the GenELinK for CCA, while the S-AppGrad is a scalable iterative algorithm for CCA with data sampling. Two separate subproblems in each iteration in terms of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ for the rgCCALin and CCALin, which are equivalent to the original subproblem in terms of $\mathbf{X}$, are solved by the stochastic variance reduced gradient method (SVRG). The SVRG runs four epochs with each running $m = n$ iterations and the step-size $\eta_x = \frac{1}{\max_i \|\widetilde{\mathbf{X}}_i\|_2^2}$ or $\eta_y = \frac{1}{\max_i \|\widetilde{\mathbf{Y}}_i\|_2^2}$, where $\widetilde{\mathbf{X}}_i$ represents the i-th column of $\widetilde{\mathbf{X}}$. The initial step-size for the rgCCALin is set to $\alpha_0 = 10^{-3}$. For the S-AppGrad, we use authors' implementation with batch size of 50 and tuned step-sizes.

For evaluation of the top-k canonical subspace pair, the following four quality measures are used:

- PCC error $= 1 - \frac{\mathrm{TCC}(\widetilde{\mathbf{X}}\mathbf{\Phi}_t, \widetilde{\mathbf{Y}}\mathbf{\Psi}_t)}{\mathrm{TCC}(\widetilde{\mathbf{X}}\mathbf{\Phi}^\star, \widetilde{\mathbf{Y}}\mathbf{\Psi}^\star)}$, where the subtrahend is called the Proportion of Correlations Captured (PCC) defined via the Total Correlations Captured (TCC) (Ma et al., 2015; Ge et al., 2016), i.e., the sum of canonical correlations between two matrices;
- $\sin^2\theta_{\max}(\mathbf{X}_t, \mathbf{U}_k)$, where
$$\mathbf{X}_t = \frac{1}{\sqrt{2}}\begin{pmatrix}\mathbf{\Phi}_t \\ \mathbf{\Psi}_t\end{pmatrix} \text{ and } \mathbf{U}_k = \frac{1}{\sqrt{2}}\begin{pmatrix}\mathbf{\Phi}^\star \\ \mathbf{\Psi}^\star\end{pmatrix};$$
- $\sin^2\theta_{\max}(\mathbf{\Phi}_t, \mathbf{\Phi}^\star)$ and $\sin^2\theta_{\max}(\mathbf{\Psi}_t, \mathbf{\Psi}^\star)$.

The ground truth $(\mathbf{\Phi}^\star, \mathbf{\Psi}^\star)$ used above is based on $(\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}})$ which can be obtained by MATLAB's svds function for evaluation purpose. Again, smaller is better for each measure above.

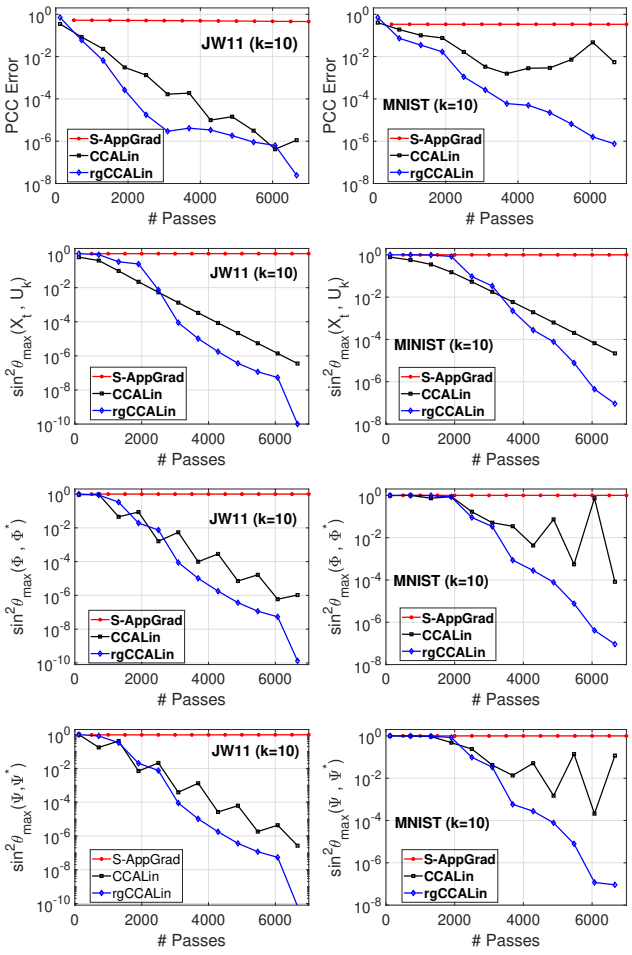

Figure 6: Performance of CCA solvers on real data.

Table 2: Summary of the CCA Data.

Data	d_x	d_y	n
JW11	273	112	30000
MNIST	392	392	60000

We test the algorithms on two CCA datasets: JW11 (Westbury, 1994) and MNIST (LeCun et al., 1998). The former is about acoustic and articulation measurements, while the latter consists of left and right halves of images. Their statistics are given in Table 2. We use $r_x = r_y = 0.1$ for both datasets and set $k = 10$. The tuned step-sizes for the S-AppGrad are 5e$-$3 and 5e$-$4 for the data, respectively. The performance of the algorithms is shown in Figure 6, where the x-axis is the number of passes over data $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$. We can see that the rgCCALin algorithm performs best in terms of all

the four measures. Contrastingly, the S-AppGrad fails to work. Note that even in the original paper on the S-AppGrad the PCC error converted from the PCC there is only in the order of 10^{-1}. Similar to the SGHA, the S-AppGrad is hard to use probably due to the sampling.

8 CONCLUSION

We study the dominant generalized eigenspace computation. Despite a large body of previous research, practical algorithms are quite lacking. In this work, we propose a practical Riemannian algorithm. It is simply based on the first-order Riemannian optimization where the inexact Riemannian gradient is used with error controlled by a least-square solver. In the meantime, the second-order information is embedded into step-sizes for both acceleration and efficiency in practice. We also extend the algorithm to CCA for the computation of the dominant canonical subspaces. Interestingly, the resulting block CCA solver does not need to use a block size of double the desired number of components. In addition, we present a novel theoretical analysis which achieves a unified linear rate of convergence by identifying the structure of the solution space, defining the structure-aware potential function, and overcoming difficulties caused by distances between subspaces of different dimensions. We conduct an extensive experimental study for examining and comparing our algorithms with baseline algorithms on synthetic and real data. Experiments consistently show that our proposed algorithm performs quite well in terms of the well-known evaluation measures. The future directions along this work include distributed computation for better scalability, incorporating preconditioning techniques into both the main problem and subproblems for acceleration, as well as trying to theoretically understand the effect of the BB step-size scheme on the performance of the proposed algorithm.

Acknowledgement

The authors sincerely thank the anonymous reviewers for their constructive comments.

References

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

Zeyuan Allen-Zhu and Yuanzhi Li. Doubly accelerated methods for faster CCA and generalized eigendecomposition. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 98–106, Sydney, Australia, 2017.

Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3174–3182, Lake Tahoe, NV, 2013.

Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 01 1988.

Kush Bhatia, Aldo Pacchiano, Nicolas Flammarion, Peter L. Bartlett, and Michael I. Jordan. Gen-oja: Simple & efficient algorithm for streaming generalized eigenvector computation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7016–7025, Montréal, Canada, 2018.

Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 129–136, Montreal, Canada, 2009.

Zhehui Chen, Xingguo Li, Lin F. Yang, Jarvis D. Haupt, and Tuo Zhao. On landscape of lagrangian functions and stochastic search for constrained nonconvex optimization. *CoRR*, abs/1806.05151, 2018.

Zhehui Chen, Xingguo Li, Lin Yang, Jarvis D. Haupt, and Tuo Zhao. On constrained nonconvex stochastic optimization: A case study for generalized eigenvalue decomposition. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 916–925, Naha, Okinawa, Japan, 2019.

R Dennis Cook and Liqiang Ni. Sufficient dimension reduction via inverse regression: A minimum discrepancy approach. *Journal of the American Statistical Association*, 100(470):410–428, 2005.

Paramveer Dhillon, Dean P Foster, and Lyle H. Ungar. Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems (NIPS)*, pages 199–207, Granada, Spain, 2011.

Jose A. Diaz-Garcia. On generalized multivariate analysis of variance. *Brazilian Journal of Probability and Statistics*, 25(1):1–13, 2011.

Rong Ge, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 2741–2750, New York, NY, 2016.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

Mario R. Guarracino, Altannar Chinchuluun, and Panos M. Pardalos. Decision rules for efficient clas-

sification of biological data. *Optimization Letters*, 3 (3):357–366, 2009.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Bruno Iannazzo and Margherita Porcelli. The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA Journal of Numerical Analysis*, 38(1):495–517, 2017.

Sham M. Kakade and Dean P. Foster. Multi-view regression via canonical correlation analysis. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT)*, pages 82–96, San Diego, CA, 2007.

Nikos Karampatziakis and Paul Mineiro. Discriminative features via generalized eigenvectors. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 494–502, Beijing, China, 2014.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Xingguo Li, Zhehui Chen, Lin Yang, Jarvis Haupt, and Tuo Zhao. Online generalized eigenvalue decomposition: Primal dual geometry and inverse-free stochastic optimization. In *the 10th NIPS Workshop on Optimization for Machine Learning*, 2017.

Zhuang Ma, Yichao Lu, and Dean P. Foster. Finding linear structure in large datasets with scalable canonical correlation analysis. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 169–178, Lille, France, 2015.

Kantilal Varichand Mardia, John T. Kent, and John M. Bibby. *Multivariate analysis*. Acad. Press, 1979.

Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, 1999.

Renaud-Alexandre Pitaval, Wei Dai, and Olav Tirkkonen. Convergence of gradient descent for low-rank matrix approximation. *IEEE Trans. Information Theory*, 61(8):4451–4457, 2015.

Marcos Raydan. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997.

Yousef Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2011.

Yousef Saad, James R. Chelikowsky, and Suzanne M. Shontz. Numerical methods for electronic structure calculations of materials. *SIAM Review*, 52(1):3–54, 2010.

Ohad Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 144–152, Lille, France, 2015.

Ohad Shamir. Fast stochastic algorithms for SVD and PCA: convergence properties and convexity. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 248–256, New York, NY, 2016.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

John R. Westbury. X-Ray microbeam speech production database user's handbook. Technical report, University of Wisconsin, Madison, 1994.

Zhiqiang Xu and Xin Gao. On truly block eigensolvers via riemannian optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 168–177, Playa Blanca, Lanzarote, Canary Islands, Spain, 2018.

Zhiqiang Xu and Ping Li. Towards practical alternating least-squares for CCA. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 14737–14746, Vancouver, Canada, 2019.

Zhiqiang Xu, Yiping Ke, and Xin Gao. A fast algorithm for matrix eigen-decompositionn. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, Sydney, Australia, 2017.

Ke Ye and Lek-Heng Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM J. Matrix Analysis Applications*, 37(3):1176–1197, 2016.

Florian Yger, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Adaptive canonical correlation analysis based on matrix manifolds. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, UK, 2012.

Zhihua Zhang. The singular value decomposition, applications and beyond. *CoRR*, abs/1510.08532, 2015.