Efficient Algorithms for Large-scale Generalized Eigenvector Computation and Canonical Correlation Analysis

Rong Ge* Chi Jin† Sham M. Kakade‡ Praneeth Netrapalli§ Aaron Sidford¶

May 30, 2016

Abstract

This paper considers the problem of canonical-correlation analysis (CCA) (Hotelling, 1936) and, more broadly, the generalized eigenvector problem for a pair of symmetric matrices. These are two fundamental problems in data analysis and scientific computing with numerous applications in machine learning and statistics (Shi and Malik, 2000; Hardoon et al., 2004; Witten et al., 2009).

We provide simple iterative algorithms, with improved runtimes, for solving these problems that are globally linearly convergent with moderate dependencies on the condition numbers and eigenvalue gaps of the matrices involved.

We obtain our results by reducing CCA to the top-k generalized eigenvector problem. We solve this problem through a general framework that simply requires black box access to an approximate linear system solver. Instantiating this framework with accelerated gradient descent we obtain a running time of $O\left(\frac{zk\sqrt{\kappa}}{\rho}\log(1/\epsilon)\log\left(k\kappa/\rho\right)\right)$ where z is the total number of nonzero entries, κ is the condition number and ρ is the relative eigenvalue gap of the appropriate matrices.

Our algorithm is linear in the input size and the number of components k up to a $\log(k)$ factor. This is essential for handling large-scale matrices that appear in practice. To the best of our knowledge this is the first such algorithm with global linear convergence. We hope that our results prompt further research and ultimately improve the practical running time for performing these important data analysis procedures on large data sets.

1 Introduction

Canonical-correlation analysis (CCA) and the generalized eigenvector problem are fundamental problems in scientific computing, data analysis, and statistics (Barnett and Preisendorfer, 1987; Friman et al., 2001).

These problems arise naturally in statistical settings. Let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times d}$ denote two large sets of data points, with empirical covariance matrices $\mathbf{S}_x = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$, $\mathbf{S}_y = \frac{1}{n}\mathbf{Y}^\top\mathbf{Y}$, and $\mathbf{S}_{xy} = \frac{1}{n}\mathbf{X}^\top\mathbf{Y}$ and suppose we wish to find features $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ that best encapsulate the similarity or dissimilarity

*Duke University. Email: rongge@cs.duke.edu
†UC Berkeley. Email: chijin@cs.berkeley.edu
‡University of Washington. Email: sham@cs.washington.edu
§Microsoft Research New England. Email: praneeth@microsoft.com
¶Microsoft Research New England. Email: asid@microsoft.com

of the data sets. CCA is the problem of maximizing the empirical correlation

$$\max_{\mathbf{x}^\top \mathbf{S}_{xx}\mathbf{x}=1 \text{ and } \mathbf{y}^\top \mathbf{S}_{yy}\mathbf{y}=1} \mathbf{x}^\top \mathbf{S}_{xy}\mathbf{y} \tag{1}$$

and thereby extracts common features of the data sets. On the other hand the generalized eigenvalue problems

$$\max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^\top \mathbf{S}_{xx}\mathbf{x}}{\mathbf{x}^\top \mathbf{S}_{yy}\mathbf{x}} \quad \text{and} \quad \max_{\mathbf{y}\neq 0} \frac{\mathbf{y}^\top \mathbf{S}_{yy}\mathbf{y}}{\mathbf{y}^\top \mathbf{S}_{xx}\mathbf{y}}$$

compute features that maximizes discrepancies between the data sets. Both these problems are easily extended to the k-feature case (See Section 3). Algorithms for solving them are commonly used to extract features to compare and contrast large data sets and are used commonly in regression (Kakade and Foster, 2007), clustering (Chaudhuri et al., 2009), classification (Karampatziakis and Mineiro, 2013), word embeddings (Dhillon et al., 2011) and more.

Despite the prevalence of these problems and the breadth of research on solving them in practice ((Barnett and Preisendorfer, 1987; Barnston and Ropelewski, 1992; Sherry and Henson, 2005; Karampatziakis and Mineiro, 2013) to name a few), there are relatively few results on obtaining provably efficient algorithms. Both problems can be reduced to performing principle component analysis (PCA), albeit on complicated matrices e.g $\mathbf{S}_{yy}^{-1/2}\mathbf{S}_{xy}^\top \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1/2}$ for CCA and $\mathbf{S}_{yy}^{-1/2}\mathbf{S}_{xx}\mathbf{S}_{yy}^{-1/2}$ for generalized eigenvector. However applying PCA to these matrices traditionally involves the formation of $\mathbf{S}_{xx}^{-1/2}$ and $\mathbf{S}_{yy}^{-1/2}$ which is prohibitive for sufficiently large datasets if we only want to estimate top-k eigenspace.

A natural open question in this area is to what degree can the formation of $\mathbf{S}_{xx}^{-1/2}$ and $\mathbf{S}_{yy}^{-1/2}$ can be bypassed to obtain efficient scalable algorithms in the case where the number of features k is much smaller than the dimensions of the problem n and d. Can we develop simple iterative practical methods that solve this problem in close to linear time when k is small and the condition number and eigenvalue gaps are bounded? While there has been recent work on solving these problems using iterative methods (Avron et al., 2014; Paul, 2015; Lu and Foster, 2014; Ma et al., 2015) we are unaware of previous provable global convergence results and more strongly, linearly convergent scalable algorithms.

The central goal of this paper is to answer this question in the affirmative. We present simple globally linearly convergent iterative methods that solve these problems. The running time of these problems scale well as the number of features and conditioning of the problem stay fixed and the size of the datasets grow. Moreover, we implement the method and perform experiments demonstrating that the techniques may be effective for large scale problems.

Specializing our results to the single feature case we show how to solve the problems all in time $O(\frac{z\sqrt{\kappa}}{\rho}\log\frac{1}{\rho}\log\frac{1}{\epsilon})$, where κ is the maximum of condition numbers of $\mathbf{S}_{xx}$ and $\mathbf{S}_{yy}$ and ρ is the eigengap of appropriate matrices and mentioned above, and z is the number of nonzero entries in $\mathbf{X}$ and $\mathbf{Y}$. To the best of our knowledge this is the first such globally linear convergent algorithm for solving these problems.

We achieve our results through a general and versatile framework that allows us to utilize fast linear system solvers in various regimes. We hope that by initiating this theoretical and practical analysis of CCA and the generalized eigenvector problem we can promote further research on the problem and ultimately advance the state-of-the-art for efficient data analysis.

1.1 Our Approach

To solve the problems motivated in the previous section we first directly reduce CCA to a generalized eigenvector problem (See Section 5). Consequently, for the majority of the paper we focus on the following:

Definition 1 (Top-k Generalized Eigenvector[1])**.** *Given symmetric matrices $\mathbf{A}, \mathbf{B}$ where $\mathbf{B}$ is positive definite compute $\mathbf{w}_1, \cdots, \mathbf{w}_k$ defined for all $i \in [k]$ by*

$$\mathbf{w}_i \in \operatorname*{argmax}_{\mathbf{w}} \left|\mathbf{w}^\top \mathbf{A}\mathbf{w}\right| \;\; s.t. \;\; \begin{matrix} \mathbf{w}^\top \mathbf{B}\mathbf{w} = 1 \text{ and} \\ \mathbf{w}^\top \mathbf{B}\mathbf{w}_j = 0 \; \forall \; j \in [i-1]. \end{matrix}$$

The generalized eigenvector is equivalent to the problem of computing the PCA of $\mathbf{A}$ in the $\mathbf{B}$ norm. Consequently, it is the same as computing the top k eigenvectors of largest absolute value of the symmetric matrix $\mathbf{M} = \mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$ and then multiplying by $\mathbf{B}^{-1/2}$.

Unfortunately, as we have discussed, explicitly computing $\mathbf{B}^{-1/2}$ is prohibitively expensive when n is large and therefore we wish to avoid forming $\mathbf{M}$ explicitly. One natural approach is to develop an iterative methods to approximately apply $\mathbf{B}^{-1/2}$ to a vector and then use that method as a subroutine to perform the power method on $\mathbf{M}$. Even if we could perform the error analysis to make this work, such an approach would likely require at least a suboptimal $\Omega(\log^2(1/\epsilon))$ iterations to achieve error ϵ.

To bypass these difficulties, we take a closer look at the power method. For some initial vector $\mathbf{x}$, let $\mathbf{y} = \mathbf{B}^{-1/2}\mathbf{M}^i\mathbf{x}$ be the result of i iterations of power method on $\mathbf{M}$ followed by multiplying $\mathbf{B}^{-1/2}$. Clearly $\mathbf{y} = (\mathbf{B}^{-1}\mathbf{A})^i\mathbf{B}^{-1/2}\mathbf{x}$. Furthermore, since we typically initialize the power method by a random vector and since $\mathbf{B}$ is positive definite, if we instead we computed $\mathbf{y} = (\mathbf{B}^{-1}\mathbf{A})^i\mathbf{x}$ for random $\mathbf{x}$ we would likely converge at the same rate as the power method at the cost of just a slightly worse initialization quality.

Consequently, we can compute our desired eigenvectors by simply alternating between applying $\mathbf{A}$ and $\mathbf{B}^{-1}$ to a random initial vector. Unfortunately, computing $\mathbf{B}^{-1}$ exactly is again outside our computational budget. At best we should only attempt to apply $\mathbf{B}^{-1}$ approximately by linear system solvers.

One of our main technical contributions is to argue about the effect of inexact solvers in this method. Whereas solving every linear system to target accuracy ϵ would again require $O(\log(1/\epsilon))$ time per linear system, which leads to a sub-optimal $O(\log^2(1/\epsilon))$ overall running time, i.e. sublinear convergence, we instead show how to warm start the linear system solvers and obtain a faster rate. We exploit the fact that as we perform many iterations of power methods, points at time t converge to eigenvectors and therefore we can initialize our linear system solver at time t carefully using our points at time $t-1$. Ultimately we show that we only need to make fixed multiplicative progress in solving the linear system in every iteration of the power method, thus the runtime for solving each linear system is independent of ϵ.

Putting these pieces together with careful error analysis yields our main result. Our algorithm only requires the ability to apply $\mathbf{A}$ to a vector and an approximate linear system solver for $\mathbf{B}$, which in turn can be obtained by just applying $\mathbf{B}$ to vectors. Consequently, our framework is versatile, scalable, and easily adaptable to take advantage of faster linear system solvers.

[1]We use the term *generalized eigenvector* to refer to a non-zero vector $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}$ for symmetric $\mathbf{A}$ and $\mathbf{B}$, not the general notion of eigenvectors for asymmetric matrices.

1.2 Previous Work

While there has been limited previous work on provably solving CCA and generalized eigenvectors, we note that there is an impressive body of literature on performing PCA(Rokhlin et al., 2009; Halko et al., 2011; Musco and Musco, 2015; Garber and Hazan, 2015; Jin et al., 2015) and solving positive semidefinite linear systems(Hestenes and Stiefel, 1952; Nesterov, 1983; Spielman and Teng, 2004). Our analysis in this paper draws on this work extensively and our results should be viewed as the principled application of them to the generalized eigenvector problem.

There has been much recent interest in designing scalable algorithms for CCA(Ma et al., 2015; Wang et al., 2015; Wang and Livescu, 2015; Michaeli et al., 2015). To our knowledge, there are no provable guarantees for approximate methods for this problem. Heuristic-based approachs (Witten et al., 2009; Lu and Foster, 2014) compute efficiently, but only give suboptimal result due to coarse approximation. The work in (Ma et al., 2015) provides one natural iterative procedure, where the per iterate computational complexity is low. This work only provides local convergence guarantees and does not provide guarantees of global convergence.

Also of note is that many recent algorithms (Ma et al., 2015; Wang et al., 2015) have mini-batch variations, but there's no guarantees for mini-batch style algorithm for CCA yet. Our algorithm can also be easily extends to a mini-batch version. While we do not explicitly analyze this variation, and we believe our analysis and techniques are helpful for extensions to this setting. We also view this as an important direction for future work.

We hope that by establishing the generalized eigenvector problem and providing provable guarantees under moderate regularity assumptions that our results may be further improved and ultimately this may advance the state-of-the-art in practical algorithms for performing data analysis.

1.3 Our Results

Our main result in this paper is a linearly convergent algorithm for computing the top generalized eigenvectors (see Definition 1). In order to be able to state our results we introduce some notation. Let $\lambda_1, \cdots, \lambda_d$ be the eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$ (their existence is guaranteed by Lemma 9 in the appendix). The eigengap $\rho \stackrel{\text{def}}{=} 1 - \frac{|\lambda_{k+1}|}{|\lambda_k|}$ and $\gamma \stackrel{\text{def}}{=} \frac{|\lambda_1|}{|\lambda_k|}$. Let z denote the number of nonzero entries in $\mathbf{A}$ and $\mathbf{B}$.

Theorem 2 (Informal version of Theorem 6)**.** *Given two matrices* $\mathbf{A}$ *and* $\mathbf{B} \in \mathbb{R}^{d\times d}$*, there is an algorithm that computes the top-k generalized eigenvectors up to an error* ϵ *in time* $\widetilde{O}(\frac{zk\sqrt{\kappa(\mathbf{B})}}{\rho}\log\frac{1}{\epsilon})$*, where* $\kappa(\mathbf{B})$ *is the condition number of* $\mathbf{B}$ *and* $\widetilde{O}(\cdot)$ *hides logarithmic terms in* d*,* γ*,* $\kappa(\mathbf{B})$ *and* ρ*, and nothing else.*

Here is a comparison of our result with previous work.

Table 1: Runtime Comparison - Generalized Eigenvectors

GenELinK(this paper)	$\widetilde{O}(\frac{d^2k\sqrt{\kappa(\mathbf{B})}}{\rho}\log\frac{1}{\epsilon})$
Fast matrix inversion	$O(d^{2.373\ldots})$

Turning to the problem of CCA, cf. (1), the relevant parameters are $\kappa \stackrel{\text{def}}{=} \max(\kappa(\mathbf{S}_{xx}), \kappa(\mathbf{S}_{yy}))$ i.e., the maximum of the condition numbers of $\mathbf{S}_{xx}$ and $\mathbf{S}_{yy}$, $\gamma \stackrel{\text{def}}{=} \frac{|\lambda_1|}{|\lambda_k|}$ where $\lambda_1, \cdots, \lambda_k$ are the eigenvalues of $\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$ in decreasing absolute value. Let z denote the number of nonzeros in $\mathbf{X}$ and $\mathbf{Y}$. Our main results are a reduction from CCA to the generalized eigenvector problem.

Theorem 3 (Informal version of Theorem 7)**.** *Given two data matrices $\mathbf{X} \in \mathbb{R}^{d_1 \times n}$ and $\mathbf{Y} \in \mathbb{R}^{d_2 \times n}$, there is an algorithm that performs top-k CCA up to an error ϵ in time $\widetilde{O}(\frac{zk\sqrt{\kappa}}{\rho} \log \frac{1}{\epsilon})$, where $d = d_1 + d_2$ and $\widetilde{O}(\cdot)$ hides logarithmic terms in d, γ, κ and ρ, and nothing else.*

Table 2 compares our result with existing results. [2]

Table 2: Runtime Comparison - CCA

This paper	$\widetilde{O}(\frac{ndk\sqrt{\kappa}}{\rho} \log \frac{1}{\epsilon})$
S-AppGrad (Ma et al., 2015)	$\widetilde{O}(\frac{ndk\kappa}{\rho^2} \log \frac{1}{\epsilon})$
Fast matrix inversion	$O(nd^{1.373\ldots})$

We should note that the actual bounds we obtain are somewhat stronger than the above informal bounds. Some of the terms in logarithm also appear only as additive terms. Finally we also give natural stochastic extensions of our algorithms where the cost of each iteration may be much smaller than the input size. The key idea behind our approach is to use an approximate linear system solver as a black box inside power method on an appropriate matrix. We show that this dependence on a linear system solver is in some sense essential. In Section 6 we show that the generalized eigenvector problem is strictly more general than the problem of solving positive semidefinite linear systems and consequently our dependence on the condition number of $\mathbf{B}$ is in some cases optimal.

Table 3: Runtime Comparison - CCA with Different Linear Solver[3]

Our result + GD	$\widetilde{O}(\frac{ndk\kappa}{\rho} \log \frac{1}{\epsilon})$
Our result + AGD	$\widetilde{O}(\frac{ndk\sqrt{\kappa}}{\rho} \log \frac{1}{\epsilon})$
Our result + SVRG	$\widetilde{O}(\frac{dk(n+\tilde{\kappa})}{\rho} \log \frac{1}{\epsilon})$
Our result + ASVRG	$\widetilde{O}(\frac{dk(n+\sqrt{n\tilde{\kappa}})}{\rho} \log \frac{1}{\epsilon})$

Subsequent to the submission of this paper, we learned of the closely related work in (Wang et al., 2016), which presents a number of additional interesting results. We think it is worthwhile to point out that our algorithm only requires black box access to any linear solver. Although the result in Theorem 3 was stated by instantiating the linear system solver by accelerated gradient descent

[2] (Ma et al., 2015) only shows local convergence for S-AppGrad. Starting within this radius of convergence requires us to already solve the problem to a high accuracy.

[3] This table was inspired by (Wang et al., 2016) in order to facilitate comparison to existing work.

(AGD), it is immediate to apply Theorem 7 and give the corresponding rates if we instantiate it by other popular algorithms, including gradient descent (GD), stochastic variance reduction (SVRG) (Johnson and Zhang, 2013), and its accelerated version (ASVRG) (Frostig et al., 2015; Lin et al., 2015). We summarize the corresponding runtime in Table 3. There $\tilde{\kappa} \stackrel{\text{def}}{=} \max\left(\frac{\max_i \|\mathbf{x}_i\|^2}{\sigma_{\min}(\mathbf{S}_{xx})}, \frac{\max_i \|\mathbf{y}_i\|^2}{\sigma_{\min}(\mathbf{S}_{yy})}\right)$, and $\mathbf{x}_i$, $\mathbf{y}_i$ are i-th column of matrix $\mathbf{X}$ and $\mathbf{Y}$. Note by definition we always have $\tilde{\kappa} \geq \kappa$. For generalized eigenvector problem, results of similar flavor as in Table 3 can also be easily derived.

Finally, we also run experiments to demonstrate the practical effectiveness of our algorithm on both small and large scale datasets.

1.4 Paper Overview

In Section 2, we present our notation. In Section 3, we formally define the problems we solve and their relevant parameters. In Section 4, we present our results for the generalized eigenvector problem. In Section 5, we present our results for the CCA problem. In Section 6 we argue that generalized eigenvector computation is as hard as linear system solving and that our dependence on $\kappa(\mathbf{B})$ is near optimal. In Section 7, we present experimental results of our algorithms on some real world data sets. Due to space limitations, proofs are deferred to the appendix.

2 Notation

We use bold capital letters $(\mathbf{A}, \mathbf{B}, \cdots)$ to denote matrices and bold lowercase letters $(\mathbf{u}, \mathbf{v}, \cdots)$ for vectors. For symmetric positive semidefinite (PSD) matrix $\mathbf{B}$, we let $\|\mathbf{u}\|_{\mathbf{B}} \stackrel{\text{def}}{=} \sqrt{\mathbf{u}^\top \mathbf{B}\mathbf{u}}$ denote the $\mathbf{B}$-norm of u and we let $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{B}} \stackrel{\text{def}}{=} \mathbf{u}^\top \mathbf{B}\mathbf{v}$ denotes the inner product of $\mathbf{u}$ and $\mathbf{v}$ in the $\mathbf{B}$-norm. We say that a matrix $\mathbf{W}$ is $\mathbf{B}$-orthonormal if $\mathbf{W}^\top \mathbf{B}\mathbf{W} = \mathbf{I}$. We let $\sigma_i(\mathbf{A})$ denotes the i^{th} largest singular value of $\mathbf{A}$, $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ denote the smallest and largest singular values of $\mathbf{A}$ respectively. Similarly we let $\lambda_i(\mathbf{A})$ refers to the i^{th} largest eigenvalue of $\mathbf{A}$ in magnitude. We let $\text{nnz}(\mathbf{A})$ denotes the number of nonzeros in $\mathbf{A}$. We also let $\kappa(\mathbf{B})$ denote the condition number of $\mathbf{B}$ (i.e., the ratio of the largest to smallest eigenvalue).

3 Problem Statement

In this section, we recall the generalized eigenvalue problem, define our error metric, and introduce all relevant parameters. Recall that the generalized eigenvalue problem is to find k vectors $\mathbf{w}_i$, $i \in [k]$ such that

$$\mathbf{w}_i \in \operatorname*{argmax}_{\mathbf{w}} \left|\mathbf{w}^\top \mathbf{A}\mathbf{w}\right| \quad \text{s.t.} \quad \begin{array}{c} \mathbf{w}^\top \mathbf{B}\mathbf{w} = 1 \text{ and} \\ \mathbf{w}^\top \mathbf{B}\mathbf{w}_j = 0 \; \forall \; j \in [i-1]. \end{array}$$

Using stationarity conditions, it can be shown that the vectors $\mathbf{w}_i$ are given by $\mathbf{w}_i = \mathbf{v}_i$, where v_i is an eigenvector of $\mathbf{B}^{-1}\mathbf{A}$ with eigenvalue λ_i such that $|\lambda_1| \geq \cdots \geq \lambda_n$. Our goal is to recover the top-k eigen space i.e., $\text{span}\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$. In order to quantify the error in estimating the eigenspace, we use largest principal angle, which is a standard notion of distance between subspaces (Golub and Van Loan, 2012).

Definition 4 (Largest principal angle)**.** *Let $\mathcal{W}$ and $\mathcal{V}$ be two k dimensional subspaces, $\mathbf{W}$ and $\mathbf{V}$ their $\mathbf{B}$-orthonormal basis respectively. The largest principal angle $\theta(\mathcal{W}, \mathcal{V})$ in the $\mathbf{B}$-norm is defined to be*

$$\theta(\mathcal{W}, \mathcal{V}) \stackrel{\text{def}}{=} \arccos\left(\sigma_{min}\left(\mathbf{V}^\top \mathbf{B}\mathbf{W}\right)\right),$$

Intuitively, the largest principal angle corresponds to the largest angle between any vector in the span of $\mathbf{W}$ and its projection onto the span of $\mathbf{V}$. In the special case where $k = 1$, the above definition reduces to our choice in the top-1 setting. Given two matrices $\mathbf{W}$ and $\mathbf{V}$, we use $\theta(\mathbf{W}, \mathbf{V})$ to denote the largest principle angle between the subspaces spanned by the columns of $\mathbf{W}$ and $\mathbf{V}$. We say that $\mathbf{W}$ achieves an error of ϵ if $\mathbf{W}^\top \mathbf{B}\mathbf{W} = \mathbf{I}$ and $\sin\theta(\mathbf{W}, \mathbf{V}) \leq \epsilon$, where $\mathbf{V}$ is the $d \times k$ matrix whose columns are $\mathbf{v}_1, \cdots, \mathbf{v}_k$. The relevant parameters for us are the eigengap, i.e. the relative difference between k^{th} and $(k+1)^{\text{th}}$ eigenvalues, $\rho \stackrel{\text{def}}{=} 1 - \frac{|\lambda_{k+1}|}{|\lambda_k|}$, and $\kappa(\mathbf{B})$, the condition number of $\mathbf{B}$.

4 Our Results

In this section, we provide our algorithms and results for solving the generalized eigenvector problem. We present our results for the special case of computing the top generalized eigenvector (Section 4.1) followed by the general case of computing the top-k generalized eigenvectors (Section 4.2). However, first we formally define a linear system solver as follows:

Linear system solver: In each of our main results (Theorems 5 and 6) we assume black box access to an approximate linear system solver. Given a PSD matrix $\mathbf{B}$, a vector $\mathbf{b}$, an initial estimate $\mathbf{u}_0$, and an error parameter δ, we require to decrease the error by a multiplicative δ, i.e. output $\mathbf{u}_1$ with $\|\mathbf{u}_1 - \mathbf{B}^{-1}\mathbf{b}\|_{\mathbf{B}}^2 \leq \delta\|\mathbf{u}_0 - \mathbf{B}^{-1}\mathbf{b}\|_{\mathbf{B}}^2$. We let $\mathcal{T}(\delta)$ denote the time needed for this operation. Since the error metric $\|\mathbf{u}_1 - \mathbf{B}^{-1}\mathbf{b}\|_{\mathbf{B}}^2$ is equivalent to function error on minimizing the convex quadratic $f(\mathbf{u}) \stackrel{\text{def}}{=} \frac{1}{2}\mathbf{u}^\top \mathbf{B}\mathbf{u} - \mathbf{u}^\top \mathbf{b}$ up to constant scaling, an approximate linear system solver is equivalent to an optimization algorithm for $f(\mathbf{u})$. We also specialize our results using Nesterov's accelerated gradient descent to state our bounds. Stating our results using linear system solver as a blackbox allows the user to choose an efficient solver depending on the structure of $\mathbf{B}$ and helps pass any improvements in linear system solvers on to the problem of generalized eigenvectors.

4.1 Top-1 Setting

Our algorithm for computing the top generalized eigenvector, GenELin is given in Algorithm 1.

The algorithm implements an approximate power method where each iteration consists of approximately multiplying a vector by $\mathbf{B}^{-1}\mathbf{A}$. In order to do this, GenELin solves a linear system in $\mathbf{B}$ and then scales the resulting vector to have unit $\mathbf{B}$-norm. Our main result states that given an oracle for solving the linear systems,[4] the number of iterations taken by Algorithm 1 to compute the top eigenvector up to an accuracy of ϵ is at most $\frac{4}{\rho}\log\frac{1}{\epsilon\cos\theta_0}$ where $\theta_0 \stackrel{\text{def}}{=} \theta(\mathbf{w}_0, \mathbf{v}_1)$.

Theorem 5. *Recall that the linear system solver takes time $\mathcal{T}(\delta)$ to reduce the error by a factor δ. Given matrices $\mathbf{A}$ and $\mathbf{B}$, GenELin (Algorithm 1) computes a vector $\mathbf{w}_T$ achieving an error of ϵ in*

[4]For example, we could use Nesterov's accelerated gradient descent, Algorithm 4

Algorithm 1 Generalized **E**igenvector via **Lin**ear System Solver (GenELin)

Input: T, symmetric matrix $\mathbf{A}$, PSD matrix $\mathbf{B}$.
Output: top generalized eigenvector $\mathbf{w}$.

$\tilde{\mathbf{w}}_0 \leftarrow$ sample uniformly from unit sphere in $\mathbb{R}^d$
$\mathbf{w}_0 \leftarrow \tilde{\mathbf{w}}_0 / \|\tilde{\mathbf{w}}_0\|_{\mathbf{B}}$
for $t = 0, \cdots, T-1$ **do**
  $\beta_t \leftarrow \mathbf{w}_t^\top \mathbf{A}\mathbf{w}_t / \mathbf{w}_t^\top \mathbf{B}\mathbf{w}_t$
  $\tilde{\mathbf{w}}_{t+1} \leftarrow \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d}[\frac{1}{2}\mathbf{w}^\top \mathbf{B}\mathbf{w} - \mathbf{w}^\top \mathbf{A}\mathbf{w}_t]$
    {Use an optimization subroutine
    with initialization $\beta_t \mathbf{w}_t$ }
  $\mathbf{w}_{t+1} \leftarrow \tilde{\mathbf{w}}_{t+1} / \|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{B}}$
end for
Return $\mathbf{w}_T$.

$T = \frac{2}{\rho} \log \frac{1}{\epsilon \cos\theta_0}$ iterations, where $\theta_0 \stackrel{\text{def}}{=} \theta(\mathbf{w}_0, \mathbf{v}_1)$. The running time of the algorithm is at most

$$O\left(\frac{1}{\rho}\left(\log\frac{1}{\cos\theta_0}\cdot\mathcal{T}\left(\frac{\rho^2\cos^2\theta_0}{16}\right) + \log\frac{1}{\epsilon}\cdot\mathcal{T}\left(\frac{\rho^2}{16}\right)\right) + \frac{1}{\rho}(\text{nnz}(\mathbf{A}) + \text{nnz}(\mathbf{B}) + d)\log\frac{1}{\epsilon\cos\theta_0}\right).$$

Furthermore, if we use Nesterov's accelerated gradient descent (Algorithm 4) to solve the linear systems in Algorithm 1, the time can be bounded as

$$O\left(\frac{\text{nnz}(\mathbf{B})\sqrt{\kappa(\mathbf{B})}}{\rho}\left(\log\frac{1}{\cos\theta_0}\log\frac{1}{\rho\cos\theta_0} + \log\frac{1}{\epsilon}\log\frac{1}{\rho}\right) + \frac{1}{\rho}\text{nnz}(\mathbf{A})\log\frac{1}{\epsilon\cos\theta_0}\right).$$

Remarks:

- Since GenELin chooses $\mathbf{w}_0$ randomly, Lemma 13 tells us that $\cos\theta_0 \geq \frac{\zeta}{\sqrt{d\kappa(\mathbf{B})}}$ with probability greater than $1-\zeta$.
- Note that GenELin exploits the sparsity of input matrices since we only need to apply them as operators.
- Depending on computational restrictions, we can also use a subset of samples in each iteration of GenELin. In some large scale learning applications using minibatches of data in each iteration helps make the method scalable while still maintaining the quality of performance.

4.2 Top-k Setting

In this section, we give an extension of our algorithm and result for computing the top-k generalized eigenvectors. Our algorithm, GenELinK is formally given as Algorithm 2.

GenELinK is a natural generalization of GenELin from the previous section. Given an initial set of vectors $\mathbf{W}_0$, the algorithm proceeds by doing approximate orthogonal iteration. Each iteration involves solving k independent linear systems[5] and orthonormalizing the iterates. The following theorem is the main result of our paper which gives runtime bounds for Algorithm 2. As before, we

[5]Similarly, as before, we could use Nesterov's accelerated gradient descent, i.e. Algorithm 4.

Algorithm 2 Generalized **E**igenvectors via **Lin**ear System Solvers-**K** (GenELinK).

Input: T, k, symmetric matrix $\mathbf{A}$, PSD matrix $\mathbf{B}$.
a subroutine $\mathrm{GS}_{\mathbf{B}}(\cdot)$ that performs Gram-Schmidt process, with inner product $\langle \cdot, \cdot \rangle_{\mathbf{B}}$.
Output: top k eigen-space $\mathbf{W} \in \mathbb{R}^{d \times k}$.

$\tilde{\mathbf{W}}_0 \leftarrow$ random $d \times k$ matrix with each entry i.i.d from $\mathcal{N}(0,1)$
$\mathbf{W}_0 \leftarrow \mathrm{GS}_{\mathbf{B}}(\tilde{\mathbf{W}}_0)$.
for $t = 0, \cdots, T-1$ **do**
  $\Gamma_t \leftarrow (\mathbf{W}_t^\top \mathbf{B} \mathbf{W}_t)^{-1} (\mathbf{W}_t^\top \mathbf{A} \mathbf{W}_t)$
  $\tilde{\mathbf{W}}_{t+1} \leftarrow \mathrm{argmin}_{\mathbf{W}} \mathrm{tr}(\frac{1}{2} \mathbf{W}^\top \mathbf{B} \mathbf{W} - \mathbf{W}^\top \mathbf{A} \mathbf{W}_t)$
    {Use an optimization subroutine
    with initialization $\mathbf{W}_t \Gamma_t$ }
  $\mathbf{W}_{t+1} \leftarrow \mathrm{GS}_{\mathbf{B}}(\tilde{\mathbf{W}}_{t+1})$
end for
Return $\mathbf{W}_T$.

assume access to a blackbox linear system solver and also give a result instantiating the theorem with Nesterov's accelerated gradient descent algorithm.

Theorem 6. *Suppose the linear system solver takes time $\mathcal{T}(\delta)$ to reduce the error by a factor δ. Given input matrices $\mathbf{A}$ and $\mathbf{B}$, GenELinK computes a $d \times k$ matrix $\mathbf{W}_T$ which is an estimate of the top generalized eigenvectors $\mathbf{V}$ with an error of ϵ i.e., $\mathbf{W}_T^\top \mathbf{B} \mathbf{W}_T = \mathbf{I}$ and $\sin\theta_T \leq \epsilon$, where $\theta_T \stackrel{\text{def}}{=} \theta(\mathbf{W}_T, \mathbf{V})$ in $T = \frac{2}{\rho} \log \frac{1}{\epsilon \cos\theta_0}$ iterations where $\theta_0 = \theta(\mathbf{W}_0, \mathbf{V})$. The run time of this algorithm is at most*

$$O\left(\frac{1}{\rho}\left(\log\frac{1}{\cos\theta_0} \cdot \mathcal{T}\left(\frac{\rho^2 \cos^4\theta_0}{64k\gamma^2}\right) + \mathcal{T}\left(\frac{\rho^2}{64k\gamma^2}\right)\log\frac{1}{\epsilon}\right) + \frac{1}{\rho}\left(\mathrm{nnz}(\mathbf{A})\,k + \mathrm{nnz}(\mathbf{B})\,k + dk^2\right)\log\frac{1}{\epsilon\cos\theta_0}\right),$$

where $\gamma \stackrel{\text{def}}{=} \frac{|\lambda_1|}{|\lambda_k|}$, $|\lambda_1| \geq \cdots \geq |\lambda_k|$ being the top-k eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$. Furthermore, if we use Nesterov's accelerated gradient descent (Algorithm 4) to solve the linear systems in Algorithm 2, the time above can be bounded as

$$O\left(\frac{\mathrm{nnz}(\mathbf{B})\,k\sqrt{\kappa(\mathbf{B})}}{\rho}\left(\log\frac{1}{\cos\theta_0}\log\frac{k\gamma}{\rho\cos\theta_0} + \log\frac{1}{\epsilon}\log\frac{k\gamma}{\rho}\right) + \frac{\left(\mathrm{nnz}(\mathbf{A})\,k + dk^2\right)}{\rho}\log\frac{1}{\epsilon\cos\theta_0}\right).$$

Remarks:

- Lemma 13 again tells us that since $\mathbf{W}_0$ is chosen to be normalized after choosing uniformly at random from the unit sphere, $\cos\theta_0 \geq \frac{\zeta}{\sqrt{dk\kappa(\mathbf{B})}}$ with probability greater than $1-\zeta$.
- This result recovers Theorem 5 as a special case, since when $k = 1$, we also have $\gamma = \frac{|\lambda_1|}{|\lambda_1|} = 1$.

5 Application to CCA

We now outline how the CCA problem can be reduced to computing generalized eigenvectors. The CCA problem is as follows. Given two sets of data points $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_2}$, let

$\mathbf{S}_x \stackrel{\text{def}}{=} \mathbf{X}^\top\mathbf{X}/n$, $\mathbf{S}_y \stackrel{\text{def}}{=} \mathbf{Y}^\top\mathbf{Y}/n$, and $\mathbf{S}_{xy} \stackrel{\text{def}}{=} \mathbf{X}^\top\mathbf{Y}/n$. We wish to find vectors $\phi_1, \cdots, \phi_k$ and $\psi_1, \cdots, \psi_k$ which are defined recursively as

$$(\phi_i, \psi_i) \in \operatorname*{argmax}_{\phi,\psi} \phi^\top \mathbf{S}_{xy} \psi$$
$$\text{s.t.} \quad \begin{array}{c} \|\phi\|_{\mathbf{S}_x} = 1 \text{ and } \phi^\top \mathbf{S}_x \phi_j = 0 \; \forall \; j \leq i-1 \\ \|\psi\|_{\mathbf{S}_y} = 1 \text{ and } \psi^\top \mathbf{S}_y \psi_j = 0 \; \forall \; j \leq i-1. \end{array}$$

where the values of $\phi_i^\top \mathbf{S}_{xy} \psi_i$ are called canonical correlations between $\mathbf{X}$ and $\mathbf{Y}$.

For reduction, we know any stationary point of this optimization problem satisfies $\mathbf{S}_{xy}\psi_i = \lambda_i \mathbf{S}_x \phi_i$, and $\mathbf{S}_{yx}\phi_i = \mu_i \mathbf{S}_y \psi_i$, where λ_i and μ_i are two constants. Combined with the constraints, we also see that $\lambda_i = \mu_i$. This can be written in matrix form as $\begin{pmatrix} 0 & \mathbf{S}_{xy} \\ \mathbf{S}_{yx} & 0 \end{pmatrix} \begin{pmatrix} \phi_i \\ \psi_i \end{pmatrix} = \lambda_i \begin{pmatrix} \mathbf{S}_x & 0 \\ 0 & \mathbf{S}_y \end{pmatrix} \begin{pmatrix} \phi_i \\ \psi_i \end{pmatrix}$. Suppose the generalized eigenvalues of the above matrices are $-\lambda_1 < -\lambda_2 < \cdots < \lambda_2 < \lambda_1$. The top $2k$-dimensional eigen-space of this generalized eigenvalue problem corresponds to the linear subspace spanned by the eigenvectors of λ_i and $-\lambda_i$, which are $\begin{pmatrix} \phi_i \\ \psi_i \end{pmatrix}, \begin{pmatrix} -\phi_i \\ \psi_i \end{pmatrix} \; \forall \; i \in [k]$. Once we solve the top-$2k$ generalized eigenvector problem for the matrices $\begin{pmatrix} 0 & \mathbf{S}_{xy} \\ \mathbf{S}_{yx} & 0 \end{pmatrix}$ and $\begin{pmatrix} \mathbf{S}_{xx} & 0 \\ 0 & \mathbf{S}_{yy} \end{pmatrix}$, we can pick any orthonormal basis that spans the output subspace and choose a random k-dimensional projection of those vectors. The formal algorithm is given in Algorithm 3. Combining this with our results for computing generalized eigenvectors, we obtain the following result.

Algorithm 3 CCA via **Lin**ear System Solvers (CCALin)

Input: T, k, data matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1}, \mathbf{Y} \in \mathbb{R}^{n \times d_2}$
Output: top k canonical subspace $\mathbf{W}_x \in \mathbb{R}^{d_1 \times k}, \mathbf{W}_y \in \mathbb{R}^{d_2 \times k}$.
$\mathbf{S}_{xx} \leftarrow \mathbf{X}^\top\mathbf{X}/n, \; \mathbf{S}_{yy} \leftarrow \mathbf{Y}^\top\mathbf{Y}/n, \; \mathbf{S}_{xy} \leftarrow \mathbf{X}^\top\mathbf{Y}/n.$
$\mathbf{A} \leftarrow \begin{pmatrix} 0 & \mathbf{S}_{xy} \\ \mathbf{S}_{xy}^\top & 0 \end{pmatrix}, \mathbf{B} \leftarrow \begin{pmatrix} \mathbf{S}_{xx} & 0 \\ 0 & \mathbf{S}_{yy} \end{pmatrix}$
$\begin{pmatrix} \bar{\mathbf{W}}_x \in \mathbb{R}^{d_1 \times 2k} \\ \bar{\mathbf{W}}_y \in \mathbb{R}^{d_2 \times 2k} \end{pmatrix} \leftarrow \text{GenELinK}(\mathbf{A}, \mathbf{B}).$
$\mathbf{U} \leftarrow 2k \times k$ random Gaussian matrix
$\tilde{\mathbf{W}}_x \leftarrow \bar{\mathbf{W}}_x \mathbf{U}.$
$\tilde{\mathbf{W}}_y \leftarrow \bar{\mathbf{W}}_y \mathbf{U}.$
$\mathbf{W}_x = \text{GS}_{\mathbf{S}_{xx}}(\tilde{\mathbf{W}}_x), \mathbf{W}_x = \text{GS}_{\mathbf{S}_{yy}}(\tilde{\mathbf{W}}_y)$
Return $\mathbf{W}_x, \mathbf{W}_y$.

Theorem 7. *Suppose the linear system solver takes time $\mathcal{T}(\delta)$ to reduce the error by a factor δ. Given inputs $\mathbf{X}$ and $\mathbf{Y}$, with probability greater than $1-\zeta$, then there is some universal constant c, so that Algorithm 3 outputs $\mathbf{W}_x$ and $\mathbf{W}_y$ such that $\sin\theta(span(\phi_i; i \in [k]), \mathbf{W}_x) \leq \epsilon$, and*

$\sin\theta(span(\psi_i; i \in [k]), \mathbf{W}_y) \leq \epsilon$, *in time*

$$O\left(\frac{1}{\rho}\left(\log\frac{d\kappa}{\zeta}\cdot\mathcal{T}\left(\frac{c\zeta^6\rho^2}{d^2k^5\kappa^2\gamma^2}\right)+\mathcal{T}\left(\frac{c\zeta^2\rho^2}{k^3\gamma^2}\right)\log\frac{1}{\epsilon}\right)+\frac{1}{\rho}\left(\mathrm{nnz}(\mathbf{X},\mathbf{Y})\,k+dk^2\right)\log\frac{d\kappa}{\zeta\epsilon}\right),$$

where $\mathrm{nnz}(\mathbf{X},\mathbf{Y}) \stackrel{\text{def}}{=} \mathrm{nnz}(\mathbf{X})+\mathrm{nnz}(\mathbf{Y})$ *and* $\kappa \stackrel{\text{def}}{=} \max(\kappa(\mathbf{S}_{xx}),\kappa(\mathbf{S}_{yy}))$ *and* $\gamma \stackrel{\text{def}}{=} \frac{\lambda_1}{\lambda_k}$. *If we use Nesterov's accelerated gradient descent (Algorithm 4) to solve the linear systems in GenELink, then the total runtime is*

$$O\left(\frac{\mathrm{nnz}(\mathbf{X},\mathbf{Y})\,k\sqrt{\kappa}}{\rho}\left(\log\frac{d\kappa}{\zeta}\log\frac{d\kappa\gamma}{\zeta\rho}+\log\frac{1}{\epsilon}\log\frac{k\gamma}{\rho}\right)+\frac{dk^2}{\rho}\log\frac{d\kappa}{\zeta\epsilon}\right),$$

Remarks:

- Note that we depend on the maximum of the condition numbers of $\mathbf{S}_{xx}$ and $\mathbf{S}_{yy}$ since the linear systems that arise in GenELinK decompose into two separate linear systems, one in $\mathbf{S}_{xx}$ and the other in $\mathbf{S}_{yy}$.
- We can also exploit sparsity in the data matrices $\mathbf{X}$ and $\mathbf{Y}$ since we only need to apply $\mathbf{S}_{xx}, \mathbf{S}_{xy}$ or $\mathbf{S}_{yy}$ only as operators, which can be done by applying $\mathbf{X}$ and $\mathbf{Y}$ in appropriate order. Exploiting sparsity is crucial for any large scale algorithm since there are many data sets (e.g., URL dataset in our experiments) where dense operations are impractical.

6 Reduction to Linear System

Here we show that solving linear systems in $\mathbf{B}$ is inherent in solving the top-k generalized eigenvector problem in the worst case and we provide evidence a $\sqrt{\kappa(\mathbf{B})}$ factor in the running time is essential for a broad class of iterative methods for the problem.

Let $\mathbf{M}$ be a symmetric positive definite matrix and suppose we wish to solve the linear system $\mathbf{Mx}=\mathbf{m}$, i.e. compute $\mathbf{x}_*$ with $\mathbf{Mx}_*=\mathbf{m}$. If we set $\mathbf{A}=\mathbf{mm}^\top$ and $\mathbf{B}=\mathbf{M}$ then

$$\operatorname*{argmax}_{\mathbf{x}^\top\mathbf{Bx}=1}\mathbf{x}^\top\mathbf{Ax}=\frac{\mathbf{B}^{-1}\mathbf{m}}{\mathbf{m}^\top\mathbf{B}^{-1}\mathbf{m}}$$

and consequently computing the top-1 generalized eigenvector yields the solution to the linear system. Therefore, the problem of computing top-k generalized eigenvectors is in general harder than the problem of solving symmetric positive definite linear systems.

Moreover, it is well known that any method which starts at $\mathbf{m}$ and iteratively applies $\mathbf{M}$ to linear combinations of the points computed so far must apply $\mathbf{M}$ at least $\Omega(\sqrt{\kappa(\mathbf{B})})$ in order to halve the error in the standard norm for the problem (Shewchuk, 1994). Consequently, methods that solve the top-1 generalized eigenvector problem by simply applying $\mathbf{A}$ and $\mathbf{B}$, which is the same as applying $\mathbf{M}$ and taking linear combinations with $\mathbf{m}$, must apply $\mathbf{M}$ at least $\Omega(\sqrt{\kappa(\mathbf{M})})$ times to achieve small error, unless they exploit more structure of $\mathbf{M}$ or the initialization.

7 Simulations

In this section, we present our experiment results performing CCA on three benchmark datasets which are summarized in Table 4. We wish to demonstrate two things via these simulations: 1)

Table 4: Summary of Datasets

Dataset	d_1	d_2	n	sparsity [6]
MNIST	392	392	6×10^4	0.19
Penn Tree Bank	10^4	10^4	5×10^5	1×10^{-4}
URL Reputation	10^5	10^5	1×10^6	5.8×10^{-5}

the behavior of CCALin verifies our theoretical result on relatively small-scale dataset, and 2) scalability of CCALin comparing it with other existing algorithms on a large-scale dataset.

Let us now specify the error metrics we use in our experiments. The first ones are the principal angles between the estimated subspaces and the true ones. Let $\mathbf{W}_x$ and $\mathbf{W}_y$ be the estimated subspaces and $\mathbf{V}_x$, $\mathbf{V}_y$ be the true canonical subspaces. We will use principle angles $\theta_x = \theta(\mathbf{W}_x, \mathbf{V}_x)$ under $\mathbf{S}_{xx}$-norm, $\theta_y = \theta(\mathbf{W}_x, \mathbf{V}_x)$ under $\mathbf{S}_{yy}$-norm and $\theta_{\mathbf{B}} = \theta\left(\begin{pmatrix} \mathbf{V}_x & 0 \\ 0 & \mathbf{V}_y \end{pmatrix}, \begin{pmatrix} \bar{\mathbf{W}}_x \\ \bar{\mathbf{W}}_y \end{pmatrix}\right)$ [7], under the $\begin{pmatrix} \mathbf{S}_{xx} & 0 \\ 0 & \mathbf{S}_{yy} \end{pmatrix}$ norm. Unfortunately, we cannot compute these error metrics for large-scale datasets since they require knowledge of the true canonical components. Instead we will use Total Correlations Captured (TCC), which is another metric widely used by practitioners, defined to be the sum of canonical correlation between two matrices. Also, Proportion of Correlations Captured is given as

$$\text{PCC} = \text{TCC}(\mathbf{X}\mathbf{W}_x, \mathbf{Y}\mathbf{W}_y)/\text{TCC}(\mathbf{X}\mathbf{V}_x, \mathbf{Y}\mathbf{V}_y)$$

For a fair comparison with other algorithms (which usually call highly optimized matrix inversion subroutines), we use number of FLOPs instead of wall clock time to measure the performance.

7.1 Small-scale Datasets

MNIST dataset(LeCun et al., 1998) consists of 60,000 handwritten digits from 0 to 9. Each digit is a image represented by 392×392 real values in [0,1]. Here CCA is performed between left half images and right half images. The data matrix is dense but the dimension is fairly small.

Penn Tree Bank (PTB) dataset comes from full Wall Street Journal Part of Penn Tree Bank which consists of 1.17 million tokens and a vocabulary size of 43k(Marcus et al., 1993), which has already been used to successfully learn the word embedding by CCA(Dhillon et al., 2011). Here, the task is to learn correlated components between two consecutive words. We only use the top 10,000 most frequent words. Each row of data matrix $\mathbf{X}$ is an indicator vector and hence it is very sparse and $\mathbf{X}^\top\mathbf{X}$ is diagonal.

Since the input matrices are very ill conditioned, we add some regularization and replace $\mathbf{S}_{xx}$ by $\mathbf{S}_{xx} + \lambda\mathbf{I}$ (and similarly with $\mathbf{S}_{yy}$). In CCALin, we run GenELinK with $k = 10$ and accelerated gradient descent (Algorithm 4 in the supplementary material) to solve the linear systems. The results are presented in Figure 1 and Figure 2.

Figure 1 shows a typical run of CCALin from random initialization on both MNIST and PTB dataset. We see although θ_x, θ_y may be even 90 degree at some point respectively, $\theta_{\mathbf{B}}$ is always

[6] Sparsity is given by $(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y}))/(nd_1 + nd_2)$.

[7] See Algorithm 3 for definition of $\bar{\mathbf{W}}_x, \bar{\mathbf{W}}_y$

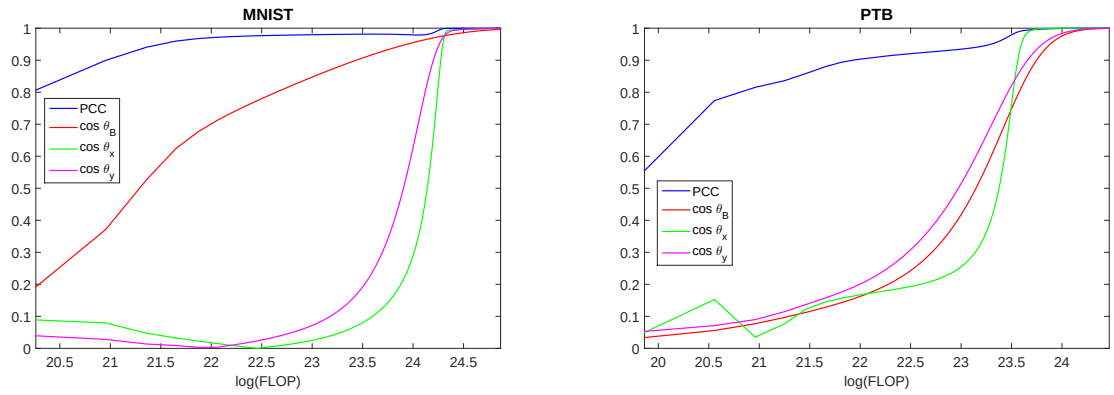

Figure 1: Global convergence of PCC and principle angles on MNIST and PTB Datasets

monotonically decreasing (as $\cos\theta_{\mathbf{B}}$ monotonically increasing) as predicted by our theory. In the end, as $\theta_{\mathbf{B}}$ goes to zero, it will push both θ_x and θ_y go to zero, and PCC go to 1. This demonstrates that our algorithm indeed converges to the true canonical space.

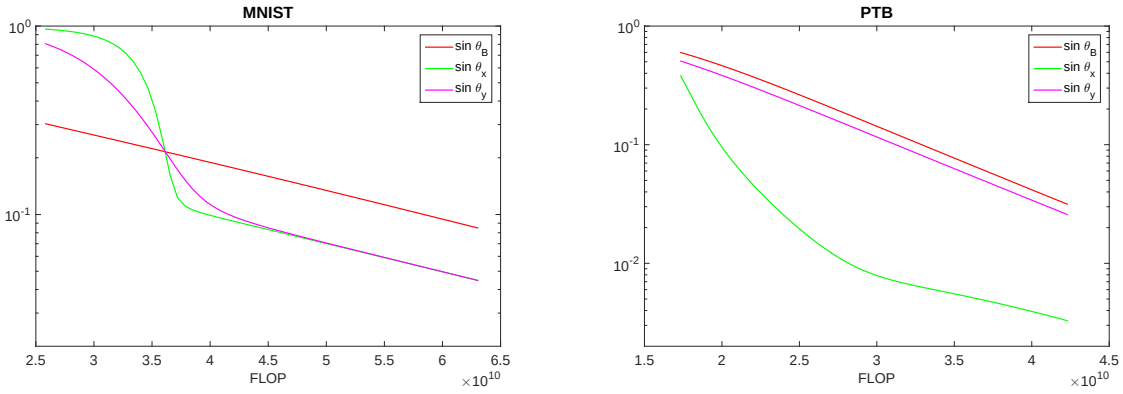

Figure 2: Linear convergence of principle angles on MNIST and PTB Datasets

Furthermore, by a more detailed examination of experimental data in Figure 1, we observe in Figure 2 that $\sin\theta_{\mathbf{B}}$ is indeed linearly convergent as we predicted in the theory. In the meantime, $\sin\theta_x$ and $\sin\theta_y$ may initially converge a bit slower than $\sin\theta_{\mathbf{B}}$, but in the end they will be upper bounded by $\sin\theta_{\mathbf{B}}$ times a constant factor, thus will eventually converge at a linear rate at least as fast as $\sin\theta_{\mathbf{B}}$.

7.2 Large-scale Dataset

URL Reputation dataset contains 2.4 million URLs and 3.2 million features including both host-based features and lexical based features. Each feature is either real valued or binary. For experiments in this section, we follow the setting of (Ma et al., 2015). We use the first 2 million samples, and run CCA between a subset of host based features and a subset of lexical based features to extract the top 20 components. Although the data matrix $\mathbf{X}$ is relatively sparse,

unlike PTB, it has strong correlations among different coordinates, which makes $\mathbf{X}^\top\mathbf{X}$ much denser ($\text{nnz}\left(\mathbf{X}^\top\mathbf{X}\right)/d_1^2 \approx 10^{-3}$).

Classical algorithms are impractical for this dataset on a typical computer, either running out of memory or requiring prohibitive amount of time. Since we cannot estimate the principal angles, we will evaluate TCC performance of CCALin.

We compare our algorithm to S-AppGrad (Ma et al., 2015) which is an iterative algorithm and PCA-CCA (Ma et al., 2015), NW-CCA (Witten et al., 2009) and DW-CCA (Lu and Foster, 2014) which are one-shot estimation procedures.

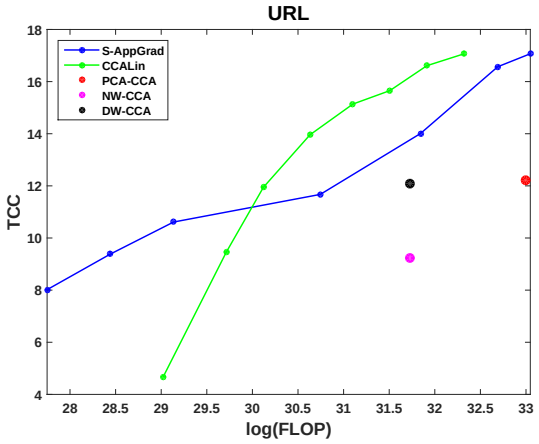

Figure 3: Comparison with existing algorithms on URL dataset

In CCALin, we employ GenELinK using stochastic accelerated gradient descent for solving linear systems using minibatches in each of the gradient steps and also leverage sparsity of the data to deal with the large data size. The result is shown in Figure 3. It is clear from the plot that our algorithm takes fewer computations than the other algorithms to achieve the same accuracy.

8 Conclusion

In summary, we have provided the first provable globally linearly convergent algorithms for solving canonical correlation analysis and the generalized eigenvector problems. We have shown that for recovering the top k components our algorithms are much faster than traditional methods based on fast matrix multiplication and singular value decomposition when $k \ll n$ and the condition numbers and eigenvalue gaps of the matrices involved are moderate. Moreover, we have provided empirical evidence that our algorithms may be useful in practice. We hope these results serve as the basis for further improvements in performing large scale data analysis both in theory and in practice.

References

Avron, H., Boutsidis, C., Toledo, S., and Zouzias, A. (2014). Efficient dimensionality reduction for canonical correlation analysis. *SIAM Journal on Scientific Computing*, 36(5):S111–S131.

Barnett, T. and Preisendorfer, R. (1987). Origins and levels of monthly and seasonal forecast skill for united states surface air temperatures determined by canonical correlation analysis. *Monthly Weather Review*, 115(9):1825–1850.

Barnston, A. G. and Ropelewski, C. F. (1992). Prediction of enso episodes using canonical correlation analysis. *Journal of climate*, 5(11):1316–1345.

Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136. ACM.

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

Friman, O., Cedefamn, J., Lundberg, P., Borga, M., and Knutsson, H. (2001). Detection of neural activity in functional mri using canonical correlation analysis. *Magnetic Resonance in Medicine*, 45(2):323–330.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. (2015). Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML2015*.

Garber, D. and Hazan, E. (2015). Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Hestenes, M. R. and Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems*, volume 49. NBS.

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

Jin, C., Kakade, S. M., Musco, C., Netrapalli, P., and Sidford, A. (2015). Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation. *CoRR*, abs/1510.08896.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS2013*, pages 315–323.

Kakade, S. M. and Foster, D. P. (2007). Multi-view regression via canonical correlation analysis. In *Learning theory*, pages 82–96. Springer.

Karampatziakis, N. and Mineiro, P. (2013). Discriminative features via generalized eigenvectors. *arXiv preprint arXiv:1310.1934*.

LeCun, Y., Cortes, C., and Burges, C. J. (1998). The mnist database of handwritten digits.

Lin, H., Mairal, J., and Harchaoui, Z. (2015). A universal catalyst for first-order optimization. *arXiv preprint arXiv:1506.02186*.

Lu, Y. and Foster, D. P. (2014). Large scale canonical correlation analysis with iterative least squares. In *Advances in Neural Information Processing Systems*, pages 91–99.

Ma, Z., Lu, Y., and Foster, D. P. (2015). Finding Linear Structure in Large Datasets with Scalable Canonical Correlation Analysis. In *Proceedings of the 32nd International Conference on Machine Learning*, JMLR Proceedings.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Michaeli, T., Wang, W., and Livescu, K. (2015). Nonparametric Canonical Correlation Analysis. *CoRR*, abs/1511.04839.

Musco, C. and Musco, C. (2015). Stronger approximate singular value decomposition via the block lanczos and power methods. *arXiv preprint arXiv:1504.05477*.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376.

Paul, S. (2015). Core-sets for canonical correlation analysis. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1887–1890. ACM.

Rokhlin, V., Szlam, A., and Tygert, M. (2009). A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124.

Rudelson, M. and Vershynin, R. (2010). Non-asymptotic theory of random matrices: extreme singular values. *arXiv preprint arXiv:1003.2990*.

Sherry, A. and Henson, R. K. (2005). Conducting and interpreting canonical correlation analysis in personality research: A user-friendly primer. *Journal of personality assessment*, 84(1):37–48.

Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

Spielman, D. A. and Teng, S.-H. (2004). Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM.

Wang, W., Arora, R., Livescu, K., and Srebro, N. (2015). Stochastic optimization for deep CCA via nonlinear orthogonal iterations. volume abs/1510.02054.

Wang, W. and Livescu, K. (2015). Large-Scale Approximate Kernel Canonical Correlation Analysis. *CoRR*, abs/1511.04773.

Wang, W., Wang, J., and Srebro, N. (2016). Globally convergent stochastic optimization for canonical correlation analysis. *arXiv preprint arXiv:1604.01870*.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, page kxp008.

A Solving Linear System via Accelerated Gradient Descent

Algorithm 4 Nesterov's accelerated gradient descent

Input: learning rate η, factor Q, initial point $\mathbf{x}_0$, T.
Output: minimizer $x^\star$ of f .
  for $t = 0, \cdots, T-1$ **do**
    $\mathbf{y}_{t+1} \leftarrow \mathbf{x}_t - (1/\beta) \cdot \nabla f(\mathbf{x}_t)$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_{t+1} + (\sqrt{Q}-1)/(\sqrt{Q}+1) \cdot (\mathbf{y}_{t+1} - \mathbf{y}_t)$
  end for
  Return $\mathbf{y}_T$.

Since we use accelerated gradient descent in our main theorems, for completeness, we put the algorithm and cite its result about iteration complexity here without proof.

Theorem 8 ((Nesterov, 1983))**.** *Let f be α-strongly convex and β-smooth, then accelerated gradient descent with learning rate $\eta = \frac{1}{\beta}$ and $Q = \beta/\alpha$ satisfies:*

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq 2(f(\mathbf{x}_0) - f(\mathbf{x}^\star)) \exp(-\frac{t}{\sqrt{Q}}) \tag{2}$$

B Proofs of Main Theorem

In this section we will prove Theorems 5, 6 and 7.

B.1 Rank-1 Setting

We first prove our claim that $\mathbf{B}^{-1}\mathbf{A}$ has an eigenbasis.

Lemma 9. *Let $(\mathbf{u}_i, \sigma_i)$ be the eigenpairs of the symmetric matrix $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$. Then $\mathbf{B}^{-1/2}\mathbf{u}_i$ is an eigenvector of $\mathbf{B}^{-1}\mathbf{A}$ with eigenvalue σ_i.*

Proof. The proof is straightforward.

$$\mathbf{B}^{-1}\mathbf{A}\left(\mathbf{B}^{-1/2}\mathbf{u}_i\right) = \mathbf{B}^{-1/2}\left(\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{u}_i\right) = \sigma_i \mathbf{B}^{-1/2}\mathbf{u}_i.$$

□

Denote the eigenpairs of $\mathbf{B}^{-1}\mathbf{A}$ by $(\lambda_i, \mathbf{v}_i)$, the above lemma further tells us that $\mathbf{v}_i^\top \mathbf{B}\mathbf{v}_j = \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$.

Recall that we defined the angle between $\mathbf{w}$ and $\mathbf{v}_1$ in the $\mathbf{B}$-norm: $\theta(\mathbf{w}, \mathbf{v}_1) = \arccos\left(|\mathbf{v}_1^\top \mathbf{B}\mathbf{w}|\right)$.

To measure the distance from optimality, we use the following potential function for normalized vector $\mathbf{w}$ ($\|\mathbf{w}\|_\mathbf{B} = 1$):

$$\tan\theta(\mathbf{w}, \mathbf{v}_1) = \frac{\sqrt{1 - |\mathbf{v}_1^\top \mathbf{B}\mathbf{w}|^2}}{|\mathbf{v}_1^\top \mathbf{B}\mathbf{w}|}. \tag{3}$$

Lemma 10. *Consider any* $\mathbf{w}$ *such that* $\|\mathbf{w}\|_{\mathbf{B}} = 1$ *and* $\tan\theta(\mathbf{w}, \mathbf{v}_1) \leq \epsilon$. *Then, we have:*

$$\cos^2\theta(\mathbf{w}, \mathbf{v}_1) = (\mathbf{v}_1^\top \mathbf{B}\mathbf{w})^2 \geq 1 - \epsilon^2 \text{ and } \mathbf{w}^\top \mathbf{A}\mathbf{w} \geq \lambda_1(1 - \epsilon^2).$$

Proof. Clearly,

$$(\mathbf{v}_1^\top \mathbf{B}\mathbf{w})^2 = \cos^2\theta(\mathbf{w}, \mathbf{v}_1) = \frac{1}{1 + \tan^2\theta(\mathbf{w}, \mathbf{v}_1)} \geq \frac{1}{1 + \epsilon^2} \geq 1 - \epsilon^2,$$

proving the first part. For the second part, we have the following:

$$\begin{aligned}\mathbf{w}^\top \mathbf{A}\mathbf{w} &= \sum_{i,j} (\mathbf{v}_i^\top \mathbf{B}\mathbf{w})(\mathbf{v}_j^\top \mathbf{B}\mathbf{w})\mathbf{v}_i^\top \mathbf{A}\mathbf{v}_j = \sum_{i,j} \lambda_j (\mathbf{v}_i^\top \mathbf{B}\mathbf{w})(\mathbf{v}_j^\top \mathbf{B}\mathbf{w})\mathbf{v}_i^\top \mathbf{B}\mathbf{v}_j \\ &= \sum_i \lambda_i (\mathbf{v}_i^\top \mathbf{B}\mathbf{w})^2 \geq \lambda_1 (\mathbf{v}_1^\top \mathbf{B}\mathbf{w})^2 \geq (1 - \epsilon^2)\lambda_1,\end{aligned}$$

proving the lemma. □

Proof of Theorem 5. We will show that the potential function $\tan\theta(\mathbf{w}_t, \mathbf{v}_1)$ decreases geometrically with t. This will directly provides an upper bound for $\sin\theta(\mathbf{w}_t, \mathbf{v}_1)$. For simplicity, through out the proof we will simply denote $\theta(\mathbf{w}_t, \mathbf{v}_i)$ as θ_t.

Recall the updates in Algorithm 1, suppose at time t, we have $\mathbf{w}_t$ such that $\|\mathbf{w}_t\|_{\mathbf{B}} = 1$. Let us say

$$\mathbf{w}_{t+1} = \frac{1}{Z}(\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t + \xi) \tag{4}$$

where Z is some normalization factor, and ξ is the error in solving the least squares. We will first prove the geometric convergence claim assuming

$$\|\xi\|_{\mathbf{B}} \leq \frac{|\lambda_1| - |\lambda_2|}{4} \min\{\cos\theta_t, \sin\theta_t\}, \tag{5}$$

and then bound the time taken by black-box linear system solver to provide such an accuracy. Since $\mathbf{w}_t$ can be written as $\mathbf{w}_t = \sum_i \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_i\right)\mathbf{v}_i$, we know $\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t = \sum_{i=1}^d \lambda_i \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_i\right)\mathbf{v}_i$. Since $\|\mathbf{w}_{t+1}\|_{\mathbf{B}} = 1$ and $\mathbf{v}_i^\top \mathbf{B}\mathbf{v}_j = \delta_{ij}$, we have

$$\begin{aligned}\tan\theta_{t+1} &= \frac{\sqrt{Z^2 - |\mathbf{v}_1^\top \mathbf{B} Z \mathbf{w}_{t+1}|^2}}{|\mathbf{v}_1^\top \mathbf{B} Z \mathbf{w}_{t+1}|} \leq \frac{\sqrt{\sum_{i=2}^d \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_i\right)^2 \lambda_i^2} + \|\xi\|_{\mathbf{B}}}{|\left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1\right)\lambda_1| - \|\xi\|_{\mathbf{B}}} \\ &\leq \frac{\sqrt{1 - \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1\right)^2}}{|\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1|} \times \frac{|\lambda_2| + \frac{\|\xi\|_{\mathbf{B}}}{\sqrt{1 - \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1\right)^2}}}{|\lambda_1| - \frac{\|\xi\|_{\mathbf{B}}}{|\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1|}} = \tan\theta_t \times \frac{|\lambda_2| + \frac{\|\xi\|_{\mathbf{B}}}{\sqrt{1 - \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1\right)^2}}}{|\lambda_1| - \frac{\|\xi\|_{\mathbf{B}}}{|\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1|}}\end{aligned}$$

By definition of θ_t, we know $\cos\theta_t = |\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1|$ and $\sin\theta_t = \sqrt{1 - \left(\mathbf{w}_t^\top \mathbf{B}\mathbf{v}_1\right)^2}$ giving us

$$\tan\theta_{t+1} \leq \tan\theta_t \times \frac{|\lambda_2| + \frac{\|\xi\|_{\mathbf{B}}}{\sin\theta_t}}{|\lambda_1| - \frac{\|\xi\|_{\mathbf{B}}}{\cos\theta_t}}.$$

Since $\|\xi\|_{\mathbf{B}} \leq \frac{|\lambda_1|-|\lambda_2|}{4}\min\{\cos\theta_t, \sin\theta_t\}$, we have that

$$\tan\theta_{t+1} \leq \frac{|\lambda_1|+3|\lambda_2|}{3|\lambda_1|+|\lambda_2|} \times \tan\theta_t.$$

Letting $\gamma = \frac{3|\lambda_1|+|\lambda_2|}{|\lambda_1|+3|\lambda_2|}$, this shows that $G(\mathbf{w}_t) \leq \gamma^t G(\mathbf{w}_0)$. Recalling the definition of eigengap $\rho = 1 - \frac{|\lambda_2|}{|\lambda_1|}$, choosing t to be

$$t \geq \frac{2}{\rho}\log\left(\frac{1}{\epsilon\cos\theta_0}\right) \geq \frac{\log\left(\frac{\tan\theta_0}{\epsilon}\right)}{\left(\frac{1}{\gamma}-1\right)} \geq \frac{\log\left(\frac{\tan\theta_0}{\epsilon}\right)}{\log\left(\frac{1}{\gamma}\right)}, \tag{6}$$

we are guaranteed that $\sin\theta_t \leq \tan\theta_t \leq \epsilon$. This number of iterations $\frac{2}{\rho}\log\left(\frac{1}{\epsilon\cos\theta_0}\right)$ could be further decompose into two phase: 1) initial phase $\frac{2}{\rho}\log\frac{1}{\cos\theta_0}$ which mainly caused by large initial angle, 2) convergence phase $\frac{2}{\rho}\log\frac{1}{\epsilon}$ which is mainly due to the high accuracy ϵ we need.

We now focus on how to obtain the iterate $\mathbf{w}_{t+1}$ using accelerated gradient descent such that the error ξ has norm bounded as in (5).

Let $f(\mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{2}\mathbf{w}^\top\mathbf{B}\mathbf{w} - \mathbf{w}^\top\mathbf{A}\mathbf{w}_t$ and recall that in each iteration, we use linear system solver to solve the following optimization problem:

$$\min_{\mathbf{w}} f(\mathbf{w}). \tag{7}$$

The minimizer of (7) is $\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t$. Define ϵ_{init} and ϵ_{des} as initial error and required destination error of linear system solver $\|\mathbf{w} - \mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t\|_{\mathbf{B}}^2$. Observe that for any $\mathbf{w}$ we have equality,

$$\|\mathbf{w} - \mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t\|_{\mathbf{B}}^2 = 2(f(\mathbf{w}) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t)) \tag{8}$$

Eq.(5) directly poses a condition on ϵ_{des}:

$$\epsilon_{\text{des}} \leq \frac{(|\lambda_1|-|\lambda_2|)^2}{16}\min\{\cos^2\theta_t, \sin^2\theta_t\}$$

Since we initialize Algorithm 4 with $\beta_t\mathbf{w}_t$, where $\beta_t \stackrel{\text{def}}{=} \frac{\mathbf{w}_t^\top\mathbf{A}\mathbf{w}_t}{\mathbf{w}_t^\top\mathbf{B}\mathbf{w}_t}$, the initial error can be bounded as follows:

$$\begin{aligned}
\epsilon_{\text{init}} &= 2(f(\beta_t\mathbf{w}_t) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t)) \\
&= 2(\min_{\beta} f(\beta\mathbf{w}_t) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t)) \leq 2(f(\lambda_1\mathbf{w}_t) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t)) \\
&= \|\lambda_1\mathbf{w}_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{w}_t\|_{\mathbf{B}}^2 \\
&= \sum_{i\geq 2}(\lambda_1-\lambda_i)^2\left(\mathbf{w}_t^\top\mathbf{B}\mathbf{v}_i\right)^2 \leq \lambda_1^2(1-\left(\mathbf{w}_t^\top\mathbf{B}\mathbf{v}_1\right)^2) = \lambda_1^2\sin^2\theta_t.
\end{aligned}$$

This means that we wish to decrease the ratio of final to initial error smaller than

$$\frac{\epsilon_{\text{des}}}{\epsilon_{\text{init}}} \leq \frac{(|\lambda_1|-|\lambda_2|)^2}{16}\min\{\cos^2\theta_t, \sin^2\theta_t\} \times \frac{1}{\lambda_1^2\sin^2\theta_t} = \frac{\rho^2}{16}\min\left\{\frac{1}{\tan^2\theta_t}, 1\right\}. \tag{9}$$

Recall we defined $\mathcal{T}(\delta)$ as the time for linear system solver to reduce the error by a factor δ. Therefore, in the initial phase where θ_t is large, it would be suffice to solve linear system up to factor $\delta = \frac{\rho^2 \cos^2\theta_0}{16} \leq \frac{\rho^2}{16\tan\theta_t}$. In convergence phase, where θ_t is small, choose $\delta = \frac{\rho^2}{16}$ would be sufficient.

Therefore, adding the computational cost of Algorithm 1 other than by linear system solver, it's not hard to get the total running time will be bounded by

$$\frac{2}{\rho}\left(\log\frac{1}{\cos\theta_0}\cdot\mathcal{T}\left(\frac{\rho^2\cos^2\theta_0}{16}\right)+\log\frac{1}{\epsilon}\cdot\mathcal{T}\left(\frac{\rho^2}{16}\right)\right)+\frac{2}{\rho}(\mathrm{nnz}\,(\mathbf{A})+\mathrm{nnz}\,(\mathbf{B})+d)\log\frac{1}{\epsilon\cos\theta_0}.$$

Furthermore, if we run Nesterov's accelerated gradient descent (Algorithm 4) on function $f(\mathbf{w})$ to solve the linear systems. Since the condition number of the optimization problem (7) is $\kappa(\mathbf{B})$, by Theorem 8, we know $\mathcal{T}(\delta) = O(\mathrm{nnz}\,(\mathbf{B})\sqrt{\kappa(\mathbf{B})}\log\frac{1}{\delta})$. Substituting this gives runtime:

$$O\left(\frac{\mathrm{nnz}\,(\mathbf{B})\sqrt{\kappa(\mathbf{B})}}{\rho}\left(\log\frac{1}{\cos\theta_0}\log\frac{1}{\rho\cos\theta_0}+\log\frac{1}{\epsilon}\log\frac{1}{\rho}\right)+\frac{1}{\rho}\mathrm{nnz}\,(\mathbf{A})\log\frac{1}{\epsilon\cos\theta_0}\right).$$

which finishes the proof. □

B.2 Top-k Setting

To prove the convergence of subspace, we need a notion of angle between subspaces. The standard definition the is principal angles.

Definition 11 (Principal angles)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be subspaces of $\mathbb{R}^d$ of dimension at least k. The principal angles $0 \leq \theta^{(1)} \leq \cdots \leq \theta^{(k)}$ between $\mathcal{X}$ and $\mathcal{Y}$ with respect to $\mathbf{B}$-based scalar product are defined recursively via:*

$$\theta^{(i)}(\mathcal{X},\mathcal{Y}) = \min\{\arccos(\frac{\langle\mathbf{x},\mathbf{y}\rangle_{\mathbf{B}}}{\|\mathbf{x}\|_{\mathbf{B}}\|\mathbf{y}\|_{\mathbf{B}}}) : \mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y},\mathbf{x}\perp_{\mathbf{B}}\mathbf{x}_j,\mathbf{y}\perp_{\mathbf{B}}\mathbf{y}_j \text{ for all } j<i\}$$
$$(\mathbf{x}_i,\mathbf{y}_i) \in \mathrm{argmin}\{\arccos(\frac{\langle\mathbf{x},\mathbf{y}\rangle_{\mathbf{B}}}{\|\mathbf{x}\|_{\mathbf{B}}\|\mathbf{y}\|_{\mathbf{B}}}) : \mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y},\mathbf{x}\perp_{\mathbf{B}}\mathbf{x}_j,\mathbf{y}\perp_{\mathbf{B}}\mathbf{y}_j \text{ for all } j<i\}$$

For matrices $\mathbf{X}$ and $\mathbf{Y}$, we use $\theta_j(\mathbf{X},\mathbf{Y})$ to denote the j-th principal angle between their range.

Since for our interest, we only care the largest principal angle, thus, in the following proof, without ambiguity, for $\mathbf{X},\mathbf{Y}\in\mathbb{R}^{d\times k}$, we use $\theta(\mathbf{X},\mathbf{Y})$ to indicate $\theta^{(k)}(\mathbf{X},\mathbf{Y})$. Next lemma will tells us this definition of $\theta(\mathbf{X},\mathbf{Y})$ to be the largest principal angle is same as what we defined in the main paper Definition 4.

Lemma 12. *Let $\mathbf{X},\mathbf{Y}\in\mathbb{R}^{d\times k}$ be orthonormal bases (w.r.t $\mathbf{B}$) for subspace $\mathcal{X},\mathcal{Y}$ respectively. Let $\mathbf{X}_\perp$ be an orthonormal basis for orthogonal complement of $\mathcal{X}$ (w.r.t $\mathbf{B}$). Then we have*

$$\cos\theta(\mathcal{X},\mathcal{Y}) = \sigma_k(\mathbf{X}^\top\mathbf{B}\mathbf{Y}),\quad \sin\theta(\mathcal{X},\mathcal{Y}) = \|\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}\| \tag{10}$$

and assuming $\mathbf{X}^\top\mathbf{B}\mathbf{Y}$ is invertible ($\theta(\mathcal{X},\mathcal{Y}) < \frac{\pi}{2}$), we have:

$$\tan\theta(\mathcal{X},\mathcal{Y}) = \|\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}\| \tag{11}$$

Proof. By definition of principal angle, it's easy to show $\cos\theta(\mathcal{X},\mathcal{Y}) = \sigma_k(\mathbf{X}^\top\mathbf{B}\mathbf{Y})$. The projection operator onto subspace $\mathcal{X}$ is $\mathbf{X}\mathbf{X}^\top\mathbf{B}$. It's also easy to show $\mathbf{X}\mathbf{X}^\top\mathbf{B} + \mathbf{X}_\perp\mathbf{X}_\perp^\top\mathbf{B} = \mathbf{I}$ Then, we have:

$$\begin{aligned}&(\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y})^\top\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y} = \mathbf{Y}^\top\mathbf{B}\mathbf{X}_\perp\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}\\ &= \mathbf{Y}^\top\mathbf{B}(\mathbf{I} - \mathbf{X}\mathbf{X}^\top\mathbf{B})\mathbf{Y} = \mathbf{Y}^\top\mathbf{B}\mathbf{Y} - (\mathbf{X}^\top\mathbf{B}\mathbf{Y})^\top(\mathbf{X}^\top\mathbf{B}\mathbf{Y}) = \mathbf{I} - (\mathbf{X}^\top\mathbf{B}\mathbf{Y})^\top(\mathbf{X}^\top\mathbf{B}\mathbf{Y})\end{aligned} \tag{12}$$

Therefore:

$$\|\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}\|^2 = 1 - \sigma_k^2(\mathbf{X}^\top\mathbf{B}\mathbf{Y}) = 1 - \cos^2\theta(\mathcal{X},\mathcal{Y}) = \sin^2\theta(\mathcal{X},\mathcal{Y}) \tag{13}$$

Similarily:

$$\begin{aligned}&[\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}]^\top\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}\\ =&[(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}]^\top[\mathbf{I} - (\mathbf{X}^\top\mathbf{B}\mathbf{Y})^\top(\mathbf{X}^\top\mathbf{B}\mathbf{Y})](\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}\\ =&[(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}]^\top(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1} - \mathbf{I}\end{aligned} \tag{14}$$

Therefore:

$$\|\mathbf{X}_\perp^\top\mathbf{B}\mathbf{Y}(\mathbf{X}^\top\mathbf{B}\mathbf{Y})^{-1}\|^2 = \frac{1}{\sigma_k^2(\mathbf{X}^\top\mathbf{B}\mathbf{Y})} - 1 = \frac{1}{\cos^2\theta(\mathcal{X},\mathcal{Y})} - 1 = \tan^2\theta(\mathcal{X},\mathcal{Y}) \tag{15}$$

Obviously, $\theta(\mathcal{X},\mathcal{Y})$ is acute, thus $\sin\theta(\mathcal{X},\mathcal{Y}) > 0$ and $\tan\theta(\mathcal{X},\mathcal{Y}) > 0$, which finishes the proof. □

Similar to the top one case, for simplicity, we denote $\theta_t \stackrel{\text{def}}{=} \theta(\mathbf{W}_t, \mathbf{V})$, where $\mathbf{V} \in \mathbb{R}^{d\times k}$ is top k eigen-vector of generalized eigenvalue problem. Now we are ready to prove the theorem. We also denote $\mathbf{V}_\perp \in \mathbb{R}^{d\times(d-k)}$. Also throughout the proof, for any matrix $\mathbf{X}$, we use notation $\|\mathbf{X}\|_\mathbf{B} \equiv \|\mathbf{B}^{\frac{1}{2}}\mathbf{X}\| \equiv \sqrt{\|\mathbf{X}^\top\mathbf{B}\mathbf{X}\|}$ and $\|\mathbf{X}\|_{\mathbf{B},F} = \|\mathbf{B}^{\frac{1}{2}}\mathbf{X}\|_F = \sqrt{\mathrm{tr}(\mathbf{X}^\top\mathbf{B}\mathbf{X})}$.

Proof of Theorem 5. Let $\mathbf{V} \in \mathbb{R}^{d\times k}, \Lambda_\mathbf{V} \in \mathbb{R}^{k\times k}$ be the top k generalized eigen-pairs; and $\mathbf{V}_\perp \in \mathbb{R}^{d\times(d-k)}, \Lambda_{\mathbf{V}_\perp} \in \mathbb{R}^{(d-k)\times(d-k)}$ be the remaining $(d-k)$ generalized eigen-pairs (assume all eigen-vectors normalized w.r.t. $\mathbf{B}$). Then, we have:

$$\begin{aligned}\mathbf{A} &= \mathbf{B}(\mathbf{V}\Lambda_\mathbf{V}\mathbf{V}^\top + \mathbf{V}_\perp\Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top)\mathbf{B}\\ \mathbf{B} &= \mathbf{B}(\mathbf{V}\mathbf{V}^\top + \mathbf{V}_\perp\mathbf{V}_\perp^\top)\mathbf{B}\end{aligned}$$

By approximately solving $\mathrm{argmin}_{\mathbf{W}\in\mathbb{R}^{d\times k}} \mathrm{tr}(\frac{1}{2}\mathbf{W}^\top\mathbf{B}\mathbf{W} - \mathbf{W}^\top\mathbf{A}\mathbf{W}_t)$ and Gram-Schmidt process, we have:

$$\mathbf{W}_{t+1} = (\mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t + \xi)\mathbf{R} \tag{16}$$

where $\mathbf{R} \in \mathbb{R}^{k\times k}$ is an invertable matrix generated by Gram-Schmidt process.

We will follow the same strategy as in top 1 case, which will first prove the geometric convergence of $\tan\theta_t$ assuming

$$\|\xi\|_\mathbf{B} \le \frac{|\lambda_k| - |\lambda_{k+1}|}{4}\min\{\sin\theta_t, \cos\theta_t\} \tag{17}$$

Note here ξ is a matrix, and $\|\xi\|_\mathbf{B} = \|\mathbf{B}^{\frac{1}{2}}\xi\| = \sqrt{\|\xi^\top\mathbf{B}\xi\|}$. Then we will bound the time taken by black-box linear system solver to provide such an accuracy.

By definition of $\tan\theta_t$ and linear algebra calculation, we have

$$\begin{aligned}
\tan\theta_{t+1} &= \|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_{t+1}(\mathbf{V}^\top \mathbf{B}\mathbf{W}_{t+1})^{-1}\| \\
&= \|\mathbf{V}_\perp^\top \mathbf{B}\tilde{\mathbf{W}}_{t+1}(\mathbf{V}^\top \mathbf{B}\tilde{\mathbf{W}}_{t+1})^{-1}\| \\
&= \|(\Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t + \mathbf{V}_\perp^\top \mathbf{B}\xi)(\Lambda_{\mathbf{V}}\mathbf{V}^\top \mathbf{B}\mathbf{W}_t + \mathbf{V}^\top \mathbf{B}\xi)^{-1}\| \\
&\le \frac{\|(\Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t + \mathbf{V}_\perp^\top \mathbf{B}\xi)(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|}{\sigma_k(\Lambda_{\mathbf{V}} + \mathbf{V}^\top \mathbf{B}\xi(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1})} \\
&\le \frac{\|\Lambda_{\mathbf{V}_\perp}\|\tan\theta_t + \|\mathbf{V}_\perp^\top \mathbf{B}\xi(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|}{\sigma_k(\Lambda_{\mathbf{V}}) - \|\mathbf{V}^\top \mathbf{B}\xi(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|} \\
&\le \frac{\|\Lambda_{\mathbf{V}_\perp}\|\tan\theta_t + \|\mathbf{V}_\perp^\top \mathbf{B}\xi\|\|\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|}{\sigma_k(\Lambda_{\mathbf{V}}) - \|\mathbf{V}^\top \mathbf{B}\xi\|\|(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|} \\
&= \frac{\|\Lambda_{\mathbf{V}_\perp}\|\tan\theta_t + \frac{\|\mathbf{V}_\perp^\top \mathbf{B}\xi\|}{\cos\theta_t}}{\sigma_k(\Lambda_{\mathbf{V}}) - \frac{\|\mathbf{V}^\top \mathbf{B}\xi\|}{\cos\theta_t}} \\
&\le \tan\theta_t \frac{|\lambda_{k+1}| + \frac{\|\xi\|_{\mathbf{B}}}{\sin\theta_t}}{|\lambda_k| - \frac{\|\xi\|_{\mathbf{B}}}{\cos\theta_t}} \qquad (18)
\end{aligned}$$

Since $\|\xi\|_{\mathbf{B}} \le \frac{|\lambda_k|-|\lambda_{k+1}|}{4}\min\{\sin\theta_t, \cos\theta_t\}$, we have that:

$$\begin{aligned}
\tan\theta_{t+1} &\le \frac{|\lambda_k| + 3|\lambda_{k+1}|}{3|\lambda_k| + |\lambda_{k+1}|}\tan\theta_t \qquad (19) \\
&= (1 - \frac{2(|\lambda_k| - |\lambda_{k+1}|)}{3|\lambda_k| + |\lambda_{k+1}|})\tan\theta_t \le \exp(-\frac{|\lambda_k| - |\lambda_{k+1}|}{2|\lambda_k|})\tan\theta_t \qquad (20)
\end{aligned}$$

Recall in this problem $\rho = 1 - \frac{|\lambda_{k+1}|}{|\lambda_k|}$, therefore, we know:

$$\sin\theta_t \le \tan\theta_t \le \exp(-\frac{\rho}{2}\cdot t)\tan\theta_0 \le \exp(-\frac{\rho}{2}\cdot t)\frac{1}{\cos\theta_0} \qquad (21)$$

If we want $\sin\theta_t \le \epsilon$, which gives iterations:

$$t \ge \frac{2}{\rho}\log\frac{1}{\epsilon\cos\theta_0} \qquad (22)$$

Let $f(\mathbf{W}) = \mathrm{tr}(\frac{1}{2}\mathbf{W}^\top \mathbf{B}\mathbf{W} - \mathbf{W}^\top \mathbf{A}\mathbf{W}_t)$. For this problem, we can view $\mathbf{W}$ as a dk dimensional vector, and use linear system to solve this d, k dimensional problem. Therefore, if we represent $\mathbf{W}$ in terms of matrix, the corresponding linear system error is $\|\mathbf{W} - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t\|_{\mathbf{B},F}$, recall $\|\mathbf{W}\|_{\mathbf{B},F} = \|\mathbf{B}^{\frac{1}{2}}\mathbf{W}\|_F = \sqrt{\mathrm{tr}(\mathbf{W}^\top \mathbf{B}\mathbf{W})}$. To satisfy the accuracy requirement, we only need

$$\epsilon_{\mathrm{des}} = \|\xi\|_{\mathbf{B},F}^2 \le \frac{(|\lambda_k| - |\lambda_{k+1}|)^2}{16}\min\{\sin^2\theta_t, \cos^2\theta_t\} \qquad (23)$$

Recall we initialize the linear system solver with $\mathbf{W}_t\Gamma_t$ with $\Gamma_t = (\mathbf{W}_t^\top \mathbf{B}\mathbf{W}_t)^{-1}(\mathbf{W}_t^\top \mathbf{A}\mathbf{W}_t)$, we then have

$$\begin{aligned}
\epsilon_{\mathrm{init}} &= \|\mathbf{W}_t\Gamma_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t\|_{\mathbf{B},F}^2 = \mathrm{tr}[(\mathbf{W}_t\Gamma_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)^\top \mathbf{B}(\mathbf{W}_t\Gamma_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)] \\
&= 2[f(\mathbf{W}_t\Gamma_t) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)] = 2[\operatorname*{argmin}_{\Gamma\in\mathbb{R}^{k\times k}} f(\mathbf{W}_t\Gamma) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)] \qquad (24)
\end{aligned}$$

Let $\hat{\Gamma}_t = (\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\Lambda_{\mathbf{V}}(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)$, and observe $\|\xi\|_{\mathbf{B},F}^2 = \|\mathbf{B}^{\frac{1}{2}}\xi\|_F^2 = \|\mathbf{V}^\top \mathbf{B}\xi\|_F^2 + \|\mathbf{V}_\perp^\top \mathbf{B}\xi\|_F^2$ (Pythagorean theorem under $\mathbf{B}$ norm), then we have:

$$\begin{aligned}
\epsilon_{\text{init}} =& \|\mathbf{W}_t\Gamma_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t\|_{\mathbf{B},F}^2 = 2[\operatorname*{argmin}_{\Gamma\in\mathbb{R}^{k\times k}} f(\mathbf{W}_t\Gamma) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)] \\
\leq& 2[f(\mathbf{W}_t\hat{\Gamma}_t) - f(\mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)] = \|\mathbf{W}_t\hat{\Gamma}_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t\|_{\mathbf{B},F}^2 \\
=& \|\mathbf{V}^\top \mathbf{B}(\mathbf{W}_t\hat{\Gamma}_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)\|_F^2 + \|\mathbf{V}_\perp^\top \mathbf{B}(\mathbf{W}_t\hat{\Gamma}_t - \mathbf{B}^{-1}\mathbf{A}\mathbf{W}_t)\|_F^2 \\
=& \|\mathbf{V}^\top \mathbf{B}\mathbf{W}_t\hat{\Gamma}_t - \Lambda_{\mathbf{V}}\mathbf{V}^\top \mathbf{B}\mathbf{W}_t\|_F^2 + \|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\hat{\Gamma}_t - \Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\|_F^2 \\
=& 0 + \|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\hat{\Gamma}_t - \Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\|_F^2 \\
\leq& k\|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\hat{\Gamma}_t - \Lambda_{\mathbf{V}_\perp}\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_t\|^2 \\
\leq& 2k\sin^2\theta_t(\|\hat{\Gamma}_t\|^2 + \|\Lambda_{\mathbf{V}_\perp}\|^2) \leq 4k|\lambda_1|^2\tan^2\theta_t
\end{aligned} \tag{25}$$

The last step is correct since $\|\Lambda_{\mathbf{V}_\perp}\| \leq |\lambda_1|$ and $\|\hat{\Gamma}_t\| \leq \|(\mathbf{V}^\top \mathbf{B}\mathbf{W}_t)^{-1}\|\|\Lambda_{\mathbf{V}}\|\|\mathbf{V}^\top \mathbf{B}^{\frac{1}{2}}\|\|\mathbf{B}^{\frac{1}{2}}\mathbf{W}_t\| \leq \frac{1}{\cos\theta_t}|\lambda_1|$

This means we wish to decrease the ratio of final to initial error smaller than:

$$\frac{\epsilon_{\text{des}}}{\epsilon_{\text{init}}} \leq \frac{\rho^2}{64k\gamma^2}\min\{\frac{1}{\cos^2\theta_t}, \frac{\sin^2\theta_t}{\cos^4\theta_t}\} \tag{26}$$

where $\gamma = \frac{|\lambda_1|}{|\lambda_k|}$. Therefore, a two phase analysis of running time depending on θ_t is large or small similar to top 1 case would gives the total runtime:

$$\frac{2}{\rho}\left(\log\frac{1}{\cos\theta_0}\cdot\mathcal{T}\left(\frac{\rho^2\cos^4\theta_0}{64k\gamma^2}\right) + \log\frac{1}{\epsilon}\cdot\mathcal{T}\left(\frac{\rho^2}{64k\gamma^2}\right)\right) + \frac{2}{\rho}\left(\text{nnz}\left(\mathbf{A}\right)k + \text{nnz}\left(\mathbf{B}\right)k + dk^2\right)\log\frac{1}{\epsilon\cos\theta_0},$$

if we are using the accelerated gradient descent to solve the linear system, we are essentially solve k disjoint optimization problem, with each problem dimension d and condition number $\kappa(\mathbf{B})$. Directly apply Theorem 8 gives runtime

$$O\left(\frac{\text{nnz}\left(\mathbf{B}\right)k\sqrt{\kappa(\mathbf{B})}}{\rho}\left(\log\frac{1}{\cos\theta_0}\log\frac{k\gamma}{\rho\cos\theta_0} + \log\frac{1}{\epsilon}\log\frac{k\gamma}{\rho}\right) + \frac{\left(\text{nnz}\left(\mathbf{A}\right)k + dk^2\right)}{\rho}\log\frac{1}{\epsilon\cos\theta_0}\right).$$

□

Finally, since both results Theorem 5 and Theorem 7 are stated in terms of initialization θ_0, here we will give probablistic guarantee for random initialization.

Lemma 13 (Random Initialization)**.** *Let top k eigen-vector be $\mathbf{V} \in \mathbb{R}^{d\times k}$, and the remaining eigen-vector be $\mathbf{V}_\perp \in \mathbb{R}^{d\times(d-k)}$. If we initialize $\mathbf{W}_0$ as in Algorithm 2, then With at least probability $1-\eta$, we have:*

$$\tan\theta_0 = \|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_0(\mathbf{V}^\top \mathbf{B}\mathbf{W}_0)^{-1}\| \leq O(\frac{\sqrt{\kappa(\mathbf{B})dk}}{\eta}) \tag{27}$$

Proof. Recall $\tilde{\mathbf{W}}$ is entry-wise sampled from standard Gaussian, and

$$\begin{aligned}
\tan\theta_0 =& \|\mathbf{V}_\perp^\top \mathbf{B}\mathbf{W}_0(\mathbf{V}^\top \mathbf{B}\mathbf{W}_0)^{-1}\| = \|\mathbf{V}_\perp^\top \mathbf{B}\tilde{\mathbf{W}}_0(\mathbf{V}^\top \mathbf{B}\tilde{\mathbf{W}}_0)^{-1}\| \leq \frac{\|\mathbf{V}_\perp^\top \mathbf{B}\tilde{\mathbf{W}}_0\|}{\sigma_k(\mathbf{V}^\top \mathbf{B}\tilde{\mathbf{W}}_0)} \\
\leq& \frac{\|\mathbf{V}_\perp^\top \mathbf{B}\tilde{\mathbf{V}}_\perp\|}{\sigma_k(\mathbf{V}^\top \mathbf{B}\tilde{\mathbf{V}})}\frac{\|\tilde{\mathbf{V}_\perp}^\top \tilde{\mathbf{W}}_0\|}{\sigma_k(\tilde{\mathbf{V}}^\top \tilde{\mathbf{W}}_0)}
\end{aligned} \tag{28}$$

Where $\tilde{\mathbf{V}}_\perp, \tilde{\mathbf{V}}$ are the right singular vectors of $\mathbf{V}_\perp^\top \mathbf{B}, \mathbf{V}^\top \mathbf{B}$ respectively. Then, we have first term:

$$\frac{\|\mathbf{V}_\perp^\top \mathbf{B}\tilde{\mathbf{V}}_\perp\|}{\sigma_k(\mathbf{V}^\top \mathbf{B}\tilde{\mathbf{V}})} = \frac{\|\mathbf{V}_\perp^\top \mathbf{B}\|}{\sigma_k(\mathbf{V}^\top \mathbf{B})} \leq \frac{\|\mathbf{V}_\perp^\top \mathbf{B}^{\frac{1}{2}}\|\|\mathbf{B}^{\frac{1}{2}}\|}{\sigma_k(\mathbf{V}^\top \mathbf{B}^{\frac{1}{2}})\sigma_{\min}(\mathbf{B}^{\frac{1}{2}})} = \kappa(\mathbf{B})^{\frac{1}{2}} \tag{29}$$

The last step is true since both $\mathbf{V}_\perp^\top \mathbf{B}^{\frac{1}{2}}$ and $\mathbf{V}^\top \mathbf{B}^{\frac{1}{2}}$ are orthonormal matrix.

For the second term, we know $\|\tilde{\mathbf{V}_\perp}^\top \tilde{\mathbf{W}}_0\| \sim O(\sqrt{d}+\sqrt{k})$ with high probability, and by equation 3.2 in (Rudelson and Vershynin, 2010) we know $\sigma_k(\tilde{\mathbf{V}}^\top \tilde{\mathbf{W}}_0) \geq \frac{\eta}{\sqrt{k}}$ with probability at least $1-\eta$, which finishes the proof. □

B.3 CCA Setting

Since our approach to CCA directly calls Algorithm 2 for solving generalized eigenvalue problem as subroutine, most of the theoretical property should be clear other than random projection step in Algorithm 3. Here, we give following lemma. The proof of Theorem 7 easily follow from the combination of this lemma and Theorem 6.

Lemma 14. *If the* $\begin{pmatrix} \bar{\mathbf{W}}_x \\ \bar{\mathbf{W}}_y \end{pmatrix}$ *as constructed in Algorithm 3 has angle at most* θ *with the true top-*$2k$ *generalized eigenspace of* $\mathbf{A}, \mathbf{B}$*, then with probability* $1-\zeta$*, both* $\mathbf{W}_x$*,* $\mathbf{W}_y$ *has angle at most* $O(k^2\theta/\zeta^2)$ *with the true top-*k *canonical space of* $\mathbf{X}, \mathbf{Y}$*.*

Proof. We will prove this for $\mathbf{W}_y$, the proof for $\mathbf{W}_x$ follows directly from same strategy.

Recall $\mathbf{B} = \begin{pmatrix} \mathbf{S}_{xx} & 0 \\ 0 & \mathbf{S}_{yy} \end{pmatrix}$. Let $\Phi \in \mathbb{R}^{d_1 \times k}$ be the true top k subspace of $\mathbf{X}$ and $\Psi \in \mathbb{R}^{d_2 \times k}$ be the true top k subspace of $\mathbf{Y}$.Then by construction we know the top $2k$ subspace should be $\frac{1}{\sqrt{2}}\begin{pmatrix} \Phi & -\Phi \\ \Psi & \Psi \end{pmatrix}$.

By properties of principal angle, we know there exists an orthonormal matrix $\mathbf{R} \in \mathbb{R}^{2k \times 2k}$ such that

$$\|\frac{1}{\sqrt{2}}\mathbf{B}^{1/2}\begin{pmatrix} \Phi & -\Phi \\ \Psi & \Psi \end{pmatrix}\mathbf{R} - \mathbf{B}^{1/2}\begin{pmatrix} \bar{\mathbf{W}}_x \\ \bar{\mathbf{W}}_y \end{pmatrix}\| \leq 2\sin\frac{\theta}{2}.$$

In particular, if we only look at the last d_2 rows, we have

$$\|\frac{1}{\sqrt{2}}\mathbf{S}_{yy}^{1/2}\begin{pmatrix} \Psi & \Psi \end{pmatrix}\mathbf{R} - \mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\| \leq 2\sin\frac{\theta}{2}.$$

Let $\mathbf{U}$ be the random Gaussian projection we used, and let $\mathbf{R}\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix}$, we know

$$\begin{aligned}\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U} &= \frac{1}{\sqrt{2}}\mathbf{S}_{yy}^{1/2}\begin{pmatrix} \Psi & \Psi \end{pmatrix}\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} + \mathbf{E} \\ &= \frac{1}{\sqrt{2}}\mathbf{S}_{yy}^{1/2}\Psi(\mathbf{U}_1+\mathbf{U}_2) + \mathbf{E},\end{aligned}$$

where $\mathbf{E}$ is the error (after multipled by random matrix $\mathbf{U}$), with $\|\mathbf{E}\| \leq O(2\sqrt{k}\sin\frac{\theta}{2}) \leq O(\sqrt{k}\theta)$.

Let $\mathbf{V} = (\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U})^\top\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U}$, the orthonormalization step gives a matrix $\mathbf{W}_y$ that is equivalent (up to rotation) to $\bar{\mathbf{W}}_y\mathbf{U}\mathbf{V}^{-1/2}$. Our goal is to show $\mathbf{V}^{-1/2} \approx ((\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2))^{-1/2}$ so we get roughly Ψ.

Note that $\Psi^\top\mathbf{S}_{yy}\Psi = \mathbf{I}$, therefore $\mathbf{V} = \frac{1}{2}(\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2) + \mathbf{E}'$ where the error $\mathbf{E}' = (\frac{1}{\sqrt{2}}\mathbf{S}_{yy}^{1/2}\Psi(\mathbf{U}_1+\mathbf{U}_2))^\top\mathbf{E} + \frac{1}{\sqrt{2}}\mathbf{E}^\top(\mathbf{S}_{yy}^{1/2}\Psi(\mathbf{U}_1+\mathbf{U}_2)) + \mathbf{E}^\top\mathbf{E})$. We know with high probability $\|\mathbf{U}_1+\mathbf{U}_2\| \le O(\sqrt{k})$, with probability at least $1-\zeta$, $\sigma_{min}(\mathbf{U}_1+\mathbf{U}_2) \ge \Omega(\zeta/\sqrt{k})$. Therefore we know $\sigma_{min}[(\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2)] \ge \Omega(\zeta^2/k)$ and $\|\mathbf{E}'\| \le O(k\theta)$. By matrix perturbation for inverse we know $\|\mathbf{V}^{-1/2} - \sqrt{2}((\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2))^{-1/2}\| \le O(k^2\theta/\zeta^2)$. Since $(\mathbf{U}_1+\mathbf{U}_2)((\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2))^{-1/2} = \mathbf{R}'$ is an orthonormal matrix, we know there's some orthonormal matrix $\mathbf{R}''$ so that:

$$
\begin{aligned}
&\|\mathbf{S}_{yy}^{1/2}\mathbf{W}_y - \mathbf{S}_{yy}^{1/2}\Psi\mathbf{R}''\| = \|\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U}\mathbf{V}^{-1/2} - \mathbf{S}_{yy}^{1/2}\Psi\mathbf{R}'\| \\
\le&\|\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U}\mathbf{V}^{-1/2} - \sqrt{2}\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U}((\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2))^{-1/2}\| \\
&+ \|\sqrt{2}\mathbf{S}_{yy}^{1/2}\bar{\mathbf{W}}_y\mathbf{U}((\mathbf{U}_1+\mathbf{U}_2)^\top(\mathbf{U}_1+\mathbf{U}_2))^{-1/2} - \mathbf{S}_{yy}^{1/2}\Psi\mathbf{R}'\| \le O(k^2\theta/\zeta^2)
\end{aligned}
$$

Therefore the angle between the $\mathbf{W}_y$ and the truth Ψ is bounded by $O(k^2\theta/\zeta^2)$.

□