On the Faster Alternating Least-Squares for CCA

Zhiqiang Xu, Ping Li
Cognitive Computing Lab
Baidu Research
No. 10 Xibeiwang East Road, Beijing 100193, China
10900 NE 8th St, Bellevue, Washington 98004, USA
{xuzhiqiang04,liping11}@baidu.com

Abstract

We study alternating least-squares (ALS) for canonical correlation analysis (CCA). Recent research shows that the alternating least-squares solver for k-CCA can be directly accelerated with momentum and prominent performance gain has been observed in practice for the resulting simple algorithm. However, despite the simplicity, it is difficult for the accelerated rate to be analyzed in theory in order to explain and match the empirical performance gain. By looking into two neighboring iterations, in this work, we propose an even simpler variant of the faster alternating least-squares solver. Instead of applying momentum to each update for acceleration, the proposed variant only leverages momentum at every other iteration and can converge at a provably faster linear rate of nearly square-root dependence on the singular value gap of the whitened cross-covariance matrix. In addition to the high consistency between theory and practice, experimental studies also show that our variant of the alternating least-squares algorithm as a block CCA solver is even more pass efficient than other variants.

1 Introduction

Canonical correlation analysis (CCA) is a statistically ubiquitous technique for finding maximally correlated components of a pair of data sources to characterize the common variability. It has been widely used in the high-dimensional data analysis including regression (Kakade and Foster, 2007), clustering (Chaudhuri et al., 2009), classification (Karampatziakis and Mineiro, 2014), and word embedding (Dhillon et al., 2011), among others. Formally, given two data matrices $\mathbf{X} \in \mathbb{R}^{d_x \times n}$ and $\mathbf{Y} \in \mathbb{R}^{d_y \times n}$, the empirical cross-covariance matrix and two empirical auto-covariance matrices can be written as follows:

$$\mathbf{C}_{xy} = \tfrac{1}{n}\mathbf{X}\mathbf{Y}^\top, \quad \mathbf{C}_{xx} = \tfrac{1}{n}\mathbf{X}\mathbf{X}^\top + r_x\mathbf{I},$$
$$\mathbf{C}_{yy} = \tfrac{1}{n}\mathbf{Y}\mathbf{Y}^\top + r_y\mathbf{I},$$

respectively, where r_x and r_y are regularization parameters for avoiding ill-conditioned matrices, and $\mathbf{I}$ represents the appropriately-sized identity matrix. The goal of the block CCA, or k-CCA where $k \geq 1$, then is to find projection matrices $\mathbf{\Phi} \in \mathbb{R}^{d_x \times k}$ and $\mathbf{\Psi} \in \mathbb{R}^{d_y \times k}$ such that the sum of the top-k correlation coefficients between $\mathbf{X}$ and $\mathbf{Y}$ is maximized after projections:

$$\max_{\mathbf{\Phi}^\top \mathbf{C}_{xx}\mathbf{\Phi} = \mathbf{\Psi}^\top \mathbf{C}_{yy}\mathbf{\Psi} = \mathbf{I}} \operatorname{tr}(\mathbf{\Phi}^\top \mathbf{C}_{xy} \mathbf{\Psi}). \tag{1}$$

The maximizer of Problem (1), denoted as $(\mathbf{U}, \mathbf{V})$, can be given by the k-SVD of the whitened empirical cross-covariance matrix $\mathbf{C} = \mathbf{C}_{xx}^{-1/2}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1/2}$. That is, if $(\overline{\mathbf{U}}, \mathbf{\Sigma}, \overline{\mathbf{V}}) \overset{k\text{-SVD}}{=\!=} \operatorname{svds}(\mathbf{C}, k)$ in MATLAB format where $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ represent the top-k left and right singular subspaces of $\mathbf{C}$, respectively, and diagonal entries of the diagonal matrix $\mathbf{\Sigma}$ are the top-k singular values of $\mathbf{C}$, then we have that

$$(\mathbf{U}, \mathbf{V}) = (\mathbf{C}_{xx}^{-\frac{1}{2}}\overline{\mathbf{U}}, \mathbf{C}_{yy}^{-\frac{1}{2}}\overline{\mathbf{V}})$$

and the maximum value is $\operatorname{tr}(\mathbf{\Sigma})$, where $(\mathbf{U}, \mathbf{V})$ is actually the top-k singular subspace pair in non-Euclidean metric $(\mathbf{C}_{xx}, \mathbf{C}_{yy})$ of the space $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, often called the top-$k$ canonical subspace pair. In the high-dimensional data analysis, both the formation (i.e., whitening of the cross-covariance matrix) and k-SVD of $\mathbf{C}$ are computationally prohibitive.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Recent research has been focusing on addressing the above issue in various settings (Yger et al., 2012; Lu and Foster, 2014; Ma et al., 2015; Ge et al., 2016; Wang et al., 2016; Allen-Zhu and Li, 2017; Gao et al., 2019; Arora et al., 2017; Chen et al., 2019; Bhatia et al., 2018; Xu and Li, 2019), by exploiting the data sparsity and leveraging stochastic optimization methods such as SGD (stochastic gradient descent, Bottou (2010)) or fast stochastic optimization methods such as SVRG (stochastic variance reduced gradient, Johnson and Zhang (2013)) and accelerated SVRG (Frostig et al., 2015; Lin et al., 2015). In this work, we follow Xu and Li (2019) to consider the block and offline setting, i.e., $k \geq 1$ and the data pair $(\mathbf{X}, \mathbf{Y})$ is ready for use. Alternating least-squares (ALS) as a block CCA solver (Ge et al., 2016) is probably most popular in this setting, by virtue of the simplicity and guarantee of convergence (Ge et al., 2016; Xu and Li, 2019). In particular, Xu and Li (2019) proposed the faster alternating least-squares (FALS) with momentum acceleration. However, despite the excellent performance in practice, the FALS has not been theoretically justified, due to lack of a theoretical convergence rate that can match better empirical performance than that of the plain ALS. This seems quite difficult to tackle directly. Nonetheless, by looking into two neighboring steps of the current FALS, we find that there actually exists an even simpler algorithm that we call *accALS*. Instead of applying momentum to every update as with the FALS, the accALS essentially leverages momentum only at every other iteration. Importantly, this change makes it amenable to a theoretical analysis that eventually yields a provably faster linear rate of nearly square-root dependence on the singular value gap of the whitened cross-covariance matrix. For practice of the proposed algorithm, two pragmatic strategies from Xu and Li (2019), i.e., coupling the update equation pair and adaptively estimating the momentum parameter, are considered. We conduct experiments on the commonly used real datasets to evaluate our practical algorithm and compare it with the latest variants of alternating least-squares for CCA. Experimental results show a high consistency between theory and practice, and the accALS is even more pass efficient than the FALS.

The rest of the paper is organized as follows. Section 2 discusses recent literature on the CCA solver. Section 3 derives our simpler variant of the faster alternating least-squares and then proceeds with its analysis. The practical implementation of the algorithm is presented in Section 4 and then followed by experimental evaluations in Section 5. The paper concludes in Section 6.

2 Related Work

To accommodate different settings, there exist various types of CCA solvers (Yger et al., 2012; Gao et al., 2019; Arora et al., 2017; Chen et al., 2019; Bhatia et al., 2018; Arora and Marinov, 2019). For readers who are not familiar with CCA, we make a succinct comparison between them. The ALS solver (Ge et al., 2016) derived from the power method is simple, easy-to-use, and theoretically guaranteed to work without acceleration. But it only works for the off-line setting where data needs be ready for use. The streaming solver (Bhatia et al., 2018) can work for the on-line setting but converge very slowly (sub-linear rate). The (SGD) solver lifted via convex relaxation (Arora et al., 2017) requires a huge per-iteration cost and consumes significantly much more memory, albeit with guarantee of low iteration complexity. The (shift-and-invert) preconditioning solver (Wang et al., 2016) requires a non-trivial estimate on singular value gap and can only deliver the solution in a recursive and sequential way via deflation when $k > 1$.

We next focus on the recently proposed off-line k-CCA algorithms. The randomized CCA algorithm for two tall and thin matrices proposed in Avron et al. (2014) first reduces the dimensionality of the matrices by random projection and then feeds them to an existing CCA algorithm. The drawback of this algorithm lies in quite a high complexity in theory and not working for large d_x and d_y. Noticing this issue, Lu and Foster (2014) decomposed the problem into a series of iterative least-squares subproblems. However, the coarse approximation leads to only sub-optimal results. Ma et al. (2015) proposed the stochastic optimization of CCA with a cheap per-iteration cost but lacks of guarantee of global convergence. Moreover, the empirical performance seems not good compared to the subsequent CCA solver like CCALin (Ge et al., 2016). Wang et al. (2016) proposed inexact alternating least squares for 1-CCA but achieved only a sub-linear convergence rate. Ge et al. (2016) considered inexact alternating least-squares for k-CCA with block size $2k$ and achieved a linear convergence rate. Using notations of Section 3.1, update equations can be written as

$$\begin{cases} \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t + \boldsymbol{\xi}_t \\ \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_t + \boldsymbol{\eta}_t \end{cases},$$

where $\mathbf{R}_{t+1}$ makes $\mathbf{\Phi}_t \in \mathbb{R}^{d_x \times 2k}$ $\mathbf{C}_{xx}$-orthonormal[1] and $\mathbf{S}_{t+1}$ makes $\mathbf{\Psi}_t \in \mathbb{R}^{d_y \times 2k}$ $\mathbf{C}_{yy}$-orthonormal for $t \geq 0$. As noted in Xu and Li (2019), it is not memory

[1] That is, $\mathbf{\Phi}_t^{\top}\mathbf{C}_{xx}\mathbf{\Phi}_t = \mathbf{I}$.

efficient enough and can be improved as:

$$\begin{cases} \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t + \boldsymbol{\xi}_t \\ \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_{t+1} + \boldsymbol{\eta}_{t+1} \end{cases}, \tag{2}$$

such that $\mathbf{\Phi}_t \in \mathbb{R}^{d_x \times k}$ and $\mathbf{\Psi}_t \in \mathbb{R}^{d_y \times k}$. Xu and Li proposed the inexact Riemannian gradient method for dominant generalized eigenspace computation and applied to k-CCA. However, the theoretical complexity depends quadratically on the eigengap. Wang et al. (2016) extended the shift-and-invert preconditioning from 1-PCA (principal component analysis) (Garber et al., 2016) to 1-CCA. Allen-Zhu and Li (2017) extended the preconditioning method to k-CCA with deflation, i.e., delivering maximally correlated components one after another instead of simultaneously. Although the preconditioning method can achieve greater performance than alternating least-squares, the sequential delivery may not be suitable for some of the downstream tasks. Also, this method is not easy to use in practice due to the non-trivial parameter estimation (e.g., singular value gap) (Wang et al., 2016), and even more so in the block case. Contrastingly, the faster alternating least-squares outperforms the plain alternating least-square without those limitations for k-CCA (Xu and Li, 2019), because it is nearly as simple as the plain alternating least-squares and can run with the adaptively estimated momentum parameter to find the approximate top-k canonical subspace pair at a time. However, as we mentioned in Section 1, the faster rate of convergence is not theoretically justified. In this work, we put forward an even simpler algorithm than the current faster alternating least-squares, with a provably faster rate. Empirically, our practical version further improves the pass efficiency over the predecessor.

3 Proposed Algorithm and Analysis

In this section, we derive a simpler and faster alternating least-squares algorithm for k-CCA to sidestep the difficulty of the analysis on the current faster alternating least-squares, and then show its theoretically faster rate of convergence.

3.1 Algorithm

Recall that on top of the vanilla (truly) alternating least-squares, i.e., Eq. (2), Xu and Li (2019) proposed the following update equations of the faster alternating least-squares for k-CCA with momentum acceleration:

$$\begin{cases} \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t - \beta\mathbf{\Phi}_{t-1} + \boldsymbol{\xi}_t \\ \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_{t+1} - \beta\mathbf{\Psi}_t + \boldsymbol{\eta}_{t+1} \end{cases},$$

where $\mathbf{\Phi}_t \in \mathbb{R}^{d_x \times k}$ is $\mathbf{C}_{xx}$-orthonormal and $\mathbf{\Psi}_t \in \mathbb{R}^{d_y \times k}$ is $\mathbf{C}_{yy}$-orthonormal for $t \geq 0$, $\mathbf{R}_{t+1} = (\widetilde{\mathbf{\Phi}}_{t+1}^{\top}\mathbf{C}_{xx}\widetilde{\mathbf{\Phi}}_{t+1})^{1/2}$ with $\widetilde{\mathbf{\Phi}}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t - \beta\mathbf{\Phi}_{t-1} + \boldsymbol{\xi}_t$, and $\mathbf{S}_{t+1} = (\widetilde{\mathbf{\Psi}}_{t+1}^{\top}\mathbf{C}_{yy}\widetilde{\mathbf{\Psi}}_{t+1})^{1/2}$ with $\widetilde{\mathbf{\Psi}}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_{t+1} - \beta\mathbf{\Psi}_t + \boldsymbol{\eta}_{t+1}$. In the above update equations, $\boldsymbol{\xi}_t$ and $\boldsymbol{\eta}_{t+1}$ are errors in approximating $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t$ and $\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_{t+1}$ by a least-squares solver, respectively. In addition, $-\beta\mathbf{\Phi}_{t-1}$ and $-\beta\mathbf{\Phi}_t$ are momentum terms with β being momentum parameter. The algorithm of this faster alternating least-squares, i.e., Algorithm 2 in Xu and Li (2019), is expected to have a theoretically faster linear convergence rate than the plain alternating least-squares, i.e., Algorithm 1 in Xu and Li (2019). However, only a local and even worse linear rate in Theorem 4.1 in Xu and Li (2019) was given that does not match the empirical performance at all. It seems quite difficult to analyze it directly in theory. Particularly, we expand the update equations for two consecutive steps and then get that

$$\begin{cases} \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\big(\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_t - \beta\mathbf{\Psi}_{t-1} \\ \qquad\qquad +\boldsymbol{\eta}_t\big)\mathbf{S}_t^{-1} - \beta\mathbf{\Phi}_{t-1} + \boldsymbol{\xi}_t \\ \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\big(\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t - \beta\mathbf{\Phi}_{t-1} \\ \qquad\qquad +\boldsymbol{\xi}_t\big)\mathbf{R}_{t+1}^{-1} - \beta\mathbf{\Psi}_t + \boldsymbol{\eta}_{t+1} \end{cases}.$$

Since the right-hand side of each equation above involves both $\mathbf{\Phi}_t$ and $\mathbf{\Psi}_t$ at one to two time steps, the analysis would be much more complicated than that of the update equations without momentum where only one term at a single time step is present (see the proof of Theorem 1 in Xu and Li (2019)). For this reason, we consider the following simpler two-step update equations instead:

$$\begin{cases} \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\big(\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_t + \boldsymbol{\xi}_t\big) \\ \qquad\qquad -\beta\mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1} + \widehat{\boldsymbol{\xi}}_t \\ \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\big(\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t + \boldsymbol{\eta}_t\big) \\ \qquad\qquad -\beta\mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1} + \widehat{\boldsymbol{\eta}}_t \end{cases}, \tag{3}$$

where $\boldsymbol{\xi}_t$ and $\widehat{\boldsymbol{\xi}}_t$ now are errors in approximating $\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_t$ and $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\widehat{\mathbf{\Psi}}_t$ with $\widehat{\mathbf{\Psi}}_t = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\mathbf{\Phi}_t + \boldsymbol{\xi}_t$, respectively. Similarly, $\boldsymbol{\eta}_t$ and $\widehat{\boldsymbol{\eta}}_t$ are the errors in approximating $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t$ and $\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\widehat{\mathbf{\Phi}}_t$ with $\widehat{\mathbf{\Phi}}_t = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t + \boldsymbol{\eta}_t$, respectively. $\mathbf{R}_{t+1}$ and $\mathbf{S}_{t+1}$ are defined the same way as before, except that now

$$\widetilde{\mathbf{\Phi}}_{t+1} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\widehat{\mathbf{\Psi}}_t - \beta\mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1} + \widehat{\boldsymbol{\xi}}_t,$$
$$\widetilde{\mathbf{\Psi}}_{t+1} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^{\top}\widehat{\mathbf{\Phi}}_t - \beta\mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1} + \widehat{\boldsymbol{\eta}}_t.$$

This means that the resulting algorithm, denoted it as accALS, alternates between the update equations of the plain alternating least-squares and those of the

Algorithm 1 accALS

1: **Input:** data matrices $(\mathbf{X}, \mathbf{Y})$, block size k, momentum parameter β, iteration number T.
2: **Output:** approximate top-k canonical subspaces $(\mathbf{\Phi}_T, \mathbf{\Psi}_T)$.
3: Set $\mathbf{\Phi}_{-1} = \mathbf{0}$, $\mathbf{\Psi}_{-1} = \mathbf{0}$, $\mathbf{\Phi}_0 = \widetilde{\mathbf{\Phi}}(\widetilde{\mathbf{\Phi}}^\top \mathbf{C}_{xx} \widetilde{\mathbf{\Phi}})^{-\frac{1}{2}}$, $\mathbf{\Psi}_0 = \widetilde{\mathbf{\Psi}}(\widetilde{\mathbf{\Psi}}^\top \mathbf{C}_{yy} \widetilde{\mathbf{\Psi}})^{-\frac{1}{2}}$, where $\widetilde{\mathbf{\Phi}}$ and $\widetilde{\mathbf{\Psi}}$ are entry-wise i.i.d. standard normal
4: **for** $t = 0, 1, \cdots, T-1$ **do**

 # Perform plain alternating least-squares updates

5: $\widehat{\mathbf{\Phi}}_t \approx \arg\min_{\mathbf{\Phi} \in \mathbb{R}^{d_x \times k}} l_t(\mathbf{\Phi})$ which starts from the initial $\mathbf{\Phi}_t(\mathbf{\Phi}_t^\top \mathbf{C}_{xx} \mathbf{\Phi}_t)^{-1}(\mathbf{\Phi}_t^\top \mathbf{C}_{xy} \mathbf{\Psi}_t)$ to approximately minimize $l_t(\mathbf{\Phi}) = \frac{1}{2n}\|\mathbf{X}^\top \mathbf{\Phi} - \mathbf{Y}^\top \mathbf{\Psi}_t\|_F^2 + \frac{r_x}{2}\|\mathbf{\Phi}\|_F^2$
6: $\widehat{\mathbf{\Psi}}_t \approx \arg\min_{\mathbf{\Psi} \in \mathbb{R}^{d_y \times k}} h_t(\mathbf{\Psi})$ which starts from the initial $\mathbf{\Psi}_t(\mathbf{\Psi}_t^\top \mathbf{C}_{yy} \mathbf{\Psi}_t)^{-1}(\mathbf{\Psi}_t^\top \mathbf{C}_{xy}^\top \mathbf{\Phi}_t)$ to approximately minimize $h_t(\mathbf{\Psi}) = \frac{1}{2n}\|\mathbf{Y}^\top \mathbf{\Psi} - \mathbf{X}^\top \mathbf{\Phi}_t\|_F^2 + \frac{r_y}{2}\|\mathbf{\Psi}\|_F^2$

 # Perform faster alternating least-squares updates

7: $\widehat{\widehat{\mathbf{\Phi}}}_t \approx \arg\min_{\mathbf{\Phi} \in \mathbb{R}^{d_x \times k}} \widehat{l}_t(\mathbf{\Phi})$ which starts from the initial $\widehat{\mathbf{\Phi}}_t(\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xx} \widehat{\mathbf{\Phi}}_t)^{-1}(\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xy} \widehat{\mathbf{\Psi}}_t)$ to approximately minimize $\widehat{l}_t(\mathbf{\Phi}) = \frac{1}{2n}\|\mathbf{X}^\top \mathbf{\Phi} - \mathbf{Y}^\top \widehat{\mathbf{\Psi}}_t\|_F^2 + \frac{r_x}{2}\|\mathbf{\Phi}\|_F^2$
8: $\mathbf{C}_{xx}$-orthonormalize $(\widehat{\widehat{\mathbf{\Phi}}}_t - \beta \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1})$ such that $(\widehat{\widehat{\mathbf{\Phi}}}_t - \beta \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1}) = \mathbf{\Phi}_{t+1}\mathbf{R}_{t+1}$, where $\mathbf{R}_{t+1} = ((\widehat{\widehat{\mathbf{\Phi}}}_t - \beta \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1})^\top \mathbf{C}_{xx} (\widehat{\widehat{\mathbf{\Phi}}}_t - \beta \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1}))^{\frac{1}{2}}$
9: $\widehat{\widehat{\mathbf{\Psi}}}_t \approx \arg\min_{\mathbf{\Psi} \in \mathbb{R}^{d_y \times k}} \widehat{h}_t(\mathbf{\Psi})$ which starts from the initial $\widehat{\mathbf{\Psi}}_t(\widehat{\mathbf{\Psi}}_t^\top \mathbf{C}_{yy} \widehat{\mathbf{\Psi}}_t)^{-1}(\widehat{\mathbf{\Psi}}_t^\top \mathbf{C}_{xy}^\top \widehat{\mathbf{\Phi}}_t)$ to approximately minimize $\widehat{h}_t(\mathbf{\Psi}) = \frac{1}{2n}\|\mathbf{Y}^\top \mathbf{\Psi} - \mathbf{X}^\top \widehat{\mathbf{\Phi}}_t\|_F^2 + \frac{r_y}{2}\|\mathbf{\Psi}\|_F^2$
10: $\mathbf{C}_{yy}$-orthonormalize $(\widehat{\widehat{\mathbf{\Psi}}}_t - \beta \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1})$ such that $(\widehat{\widehat{\mathbf{\Psi}}}_t - \beta \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1}) = \mathbf{\Psi}_{t+1}\mathbf{S}_{t+1}$, where $\mathbf{S}_{t+1} = ((\widehat{\widehat{\mathbf{\Psi}}}_t - \beta \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1})^\top \mathbf{C}_{yy} (\widehat{\widehat{\mathbf{\Psi}}}_t - \beta \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1}))^{\frac{1}{2}}$
11: **end for**

faster alternating least-squares. We next elaborate on the proposed algorithm.

The pseudo code of the accALS algorithm is described in Algorithm 1. It starts from initializing $\mathbf{\Phi}_t$ and $\mathbf{\Psi}_t$ for the first two steps $t = -1, 0$ in Line 3. Particularly, $\mathbf{\Phi}_{-1} = \mathbf{0}, \mathbf{\Psi}_0 = \mathbf{0}$, and $\mathbf{\Phi}_0$ is obtained from $\mathbf{C}_{xx}$-orthonormalizing an entry-wise standard normal matrix $\widetilde{\mathbf{\Phi}}$ of size $d_x \times k$. The case of $\mathbf{\Psi}_0$ is similar. Each step[2] of Algorithm 1 first performs the plain alternating least-squares updates: $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t$ is approximated, in Line 5, with $\widehat{\mathbf{\Phi}}_t = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t + \boldsymbol{\eta}_t$ by a least-squares solver; $\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top \mathbf{\Phi}_t$ is approximated, in Line 6, by $\widehat{\mathbf{\Psi}}_t = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top \mathbf{\Phi}_t + \boldsymbol{\xi}_t$ similarly. In general, a few iterations suffice for the least-squares solver. The faster alternating least-squares updates are then performed, starting from approximating $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\widehat{\mathbf{\Psi}}_t$ with $\widehat{\widehat{\mathbf{\Phi}}}_t = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\widehat{\mathbf{\Psi}}_t + \widehat{\boldsymbol{\xi}}_t$ by a least-squares solver in Line 7. Line 8 proceeds to $\mathbf{C}_{xx}$-orthonormalize $\widetilde{\mathbf{\Phi}}_{t+1} = \widehat{\widehat{\mathbf{\Phi}}}_t - \beta \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1}$ and get $\mathbf{C}_{xx}$-orthonormal $\mathbf{\Phi}_{t+1}$. Likewise, $\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top \widehat{\mathbf{\Phi}}_t$ is approximated in Line 9 with $\widehat{\widehat{\mathbf{\Psi}}}_t = \mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top \widehat{\mathbf{\Phi}}_t + \widehat{\boldsymbol{\eta}}_t$ by a least-squares solver, followed by $\mathbf{C}_{yy}$-orthonormalizing $\widetilde{\mathbf{\Psi}}_{t+1} = \widehat{\widehat{\mathbf{\Psi}}}_t - \beta \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1}$ and getting $\mathbf{C}_{yy}$-orthonormal $\mathbf{\Psi}_{t+1}$ in Line 10.

3.2 Analysis

Before proceeding to analysis, a few definitions are introduced.

Notions and notations The ground-truth canonical subspaces $\mathbf{U}$ and $\mathbf{V}$ are the top-k left and right singular subspaces[3] of $\mathbf{C}$ in non-Euclidean metrics $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$, respectively, corresponding to $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_k)$, where σ_i, $i = 1, \cdots, \mathrm{rank}(\mathbf{C})$, are indexed in descending order, i.e., $\sigma_i \geq \sigma_j$ for $1 \leq i < j \leq \mathrm{rank}(\mathbf{C})$. Note that $\mathbf{U}$ and $\mathbf{V}$ are $\mathbf{C}_{xx}$-orthonormal and $\mathbf{C}_{yy}$-orthonormal, respectively. Let $\theta_{\max}(\mathbf{\Phi}, \mathbf{U})$ and $\theta_{\min}(\mathbf{\Phi}, \mathbf{U})$ represent the largest and smallest principal angle between the subspace spanned by $\mathbf{\Phi}$ and the subspace spanned by $\mathbf{U}$, respectively, in

[2] It corresponds to two steps of the algorithms in Xu and Li (2019)

[3] For brevity, a subspace and one of its bases share the notation throughout the paper.

the underlying metric $\mathbf{C}_{xx}$, i.e.,

$$\theta_{\max}(\mathbf{\Phi},\mathbf{U}) = cos^{-1}(\sigma_{\min}(\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{C}_{xx}\mathbf{\Phi})^{-\frac{1}{2}})),$$
$$\theta_{\min}(\mathbf{\Phi},\mathbf{U}) = cos^{-1}(\sigma_{\max}(\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{C}_{xx}\mathbf{\Phi})^{-\frac{1}{2}})),$$

where $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ represent the maximum and minimum singular value of a matrix, respectively. Note that the largest principal angle is commonly used to gauge the distance between subspaces. Define

$$\theta_t \triangleq \max\{\theta_{\max}(\mathbf{\Phi}_t,\mathbf{U}),\theta_{\max}(\mathbf{\Psi}_t,\mathbf{V})\}$$

and let $\mathrm{nnz}(\mathbf{X})$ represent the number of nonzero entries in $\mathbf{X}$ and $\kappa(\mathbf{C}_{xx})$ the condition number of $\mathbf{C}_{xx}$. The analysis will depend on the following matrices:

$$\mathbf{A}_\phi = \begin{pmatrix} \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top & -\beta\mathbf{C}_{xx} \\ \mathbf{C}_{xx} & \mathbf{0} \end{pmatrix},$$
$$\mathbf{A}_\psi = \begin{pmatrix} \mathbf{C}_{xy}^\top\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy} & -\beta\mathbf{C}_{yy} \\ \mathbf{C}_{yy} & \mathbf{0} \end{pmatrix},$$
$$\mathbf{B}_\phi = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{xx} \end{pmatrix}, \quad \mathbf{B}_\psi = \begin{pmatrix} \mathbf{C}_{yy} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix},$$
$$\mathbf{P}_t = \begin{pmatrix} \mathbf{\Phi}_t \\ \mathbf{\Phi}_{t-1}\mathbf{R}_t^{-1} \end{pmatrix}\widetilde{\mathbf{R}}_t, \quad \mathbf{Q}_t = \begin{pmatrix} \mathbf{\Psi}_t \\ \mathbf{\Psi}_{t-1}\mathbf{S}_t^{-1} \end{pmatrix}\widetilde{\mathbf{S}}_t,$$

where

$$\widetilde{\mathbf{R}}_t = \begin{cases} (\mathbf{I}+\mathbf{R}_t^{-\top}\mathbf{R}_t^{-1})^{-\frac{1}{2}}, & t>0 \\ \mathbf{I}, & t=0 \end{cases},$$
$$\widetilde{\mathbf{S}}_t = \begin{cases} (\mathbf{I}+\mathbf{S}_t^{-\top}\mathbf{S}_t^{-1})^{-\frac{1}{2}}, & t>0 \\ \mathbf{I}, & t=0 \end{cases}.$$

Moreover, let $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ represent the top-k generalized eigenspaces of matrix pairs, $(\mathbf{A}_\phi,\mathbf{B}_\phi)$ and $(\mathbf{A}_\psi,\mathbf{B}_\psi)$, respectively.

We then have the following theorem for Algorithm 1.

Theorem 3.1 *Given data matrices $(\mathbf{X},\mathbf{Y})\in\mathbb{R}^{d_x\times n}\times\mathbb{R}^{d_y\times n}$, if $\sigma_k^2 > 2\sqrt{\beta} = \sigma_{k+1}^2$, Algorithm 1 then computes $\mathbf{\Phi}_T$ and $\mathbf{\Psi}_T$ which are estimates of top-k canonical subspaces $(\mathbf{U},\mathbf{V})$ such that $\sin\theta_T \le \epsilon$ and $\mathbf{\Phi}_T^\top\mathbf{C}_{xx}\mathbf{\Phi}_T = \mathbf{\Psi}_T^\top\mathbf{C}_{yy}\mathbf{\Psi}_T = \mathbf{I}$, in $T = O(\sqrt{\frac{\sigma_k^2}{\sigma_k^2-\sigma_{k+1}^2}}\log\frac{1}{\epsilon(\sigma_k^2-\sigma_{k+1}^2)\cos\theta_0})$ iterations. If Nesterov's accelerated gradient descent is used as the least-squares solver, the overall running time is at most*

$$O\Big(\big(dk^2+k\mathrm{nnz}(\mathbf{X},\mathbf{Y})\kappa^{\frac{1}{2}}(\mathbf{X},\mathbf{Y})\log\tfrac{c_1c_2}{(\sigma_k^2-\sigma_{k+1}^2)\cos\theta_0}\big)\sqrt{\tfrac{\sigma_k^2}{\sigma_k^2-\sigma_{k+1}^2}}\log\tfrac{1}{\epsilon(\sigma_k^2-\sigma_{k+1}^2)\cos\theta_0}\Big),$$

where $d = \max\{d_x,d_y\}$, $\mathrm{nnz}(\mathbf{X},\mathbf{Y}) = \mathrm{nnz}(\mathbf{X}) + \mathrm{nnz}(\mathbf{Y})$, $\kappa(\mathbf{X},\mathbf{Y}) = \max\{\kappa(\mathbf{C}_{xx}),\kappa(\mathbf{C}_{yy})\}$, and

$$c_1 = \max_t \frac{\tan\max\{\theta_t,\,\theta_{\max}(\widehat{\mathbf{\Phi}}_t,\mathbf{U}),\,\theta_{\max}(\widehat{\mathbf{\Psi}}_t,\mathbf{V})\}}{\tan\min\{\theta_{\max}(\mathbf{P}_t,\widetilde{\mathbf{U}}),\,\theta_{\max}(\mathbf{Q}_t,\widetilde{\mathbf{V}})\}},$$
$$c_2 = \max\{\max_t\tfrac{\theta_{\max}(\mathbf{P}_t,\widetilde{\mathbf{U}})}{\theta_{\min}(\mathbf{P}_t,\widetilde{\mathbf{U}})},\max_t\tfrac{\theta_{\max}(\mathbf{Q}_t,\widetilde{\mathbf{V}})}{\theta_{\min}(\mathbf{Q}_t,\widetilde{\mathbf{V}})}\}.$$

Proof Sketch We shall sketch the proof idea of the theorem here, and have the complete and long proof deferred to the supplementary material. Note that Algorithm 1 is induced by the proposed update (3). It is easy to check that update (3) can be rewritten as two equivalent equations in those previously defined matrices, i.e.,

$$\begin{cases} \mathbf{P}_{t+1}\widetilde{\mathbf{R}}_{t+1}^{-1}\mathbf{R}_{t+1}\widetilde{\mathbf{R}}_t = \mathbf{B}_\phi^{-1}\mathbf{A}_\phi\mathbf{P}_t + \boldsymbol{\delta}_t, \\ \mathbf{Q}_{t+1}\widetilde{\mathbf{S}}_{t+1}^{-1}\mathbf{S}_{t+1}\widetilde{\mathbf{S}}_t = \mathbf{B}_\psi^{-1}\mathbf{A}_\psi\mathbf{Q}_t + \boldsymbol{\rho}_t, \end{cases}$$

where $\boldsymbol{\delta}_t = ((\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\boldsymbol{\xi}_t+\widehat{\boldsymbol{\xi}}_t)^\top,\mathbf{0}^\top)^\top\widetilde{\mathbf{R}}_t$ and $\boldsymbol{\rho}_t = ((\mathbf{C}_{yy}^{-1}\mathbf{C}_{xy}^\top\boldsymbol{\eta}_t+\widehat{\boldsymbol{\eta}}_t)^\top,\mathbf{0}^\top)^\top\widetilde{\mathbf{S}}_t$. This boils down to running the inexact power method on the augmented matrix pairs $(\mathbf{A}_\phi,\mathbf{B}_\phi)$ and $(\mathbf{A}_\psi,\mathbf{B}_\psi)$ with errors $\boldsymbol{\delta}_t$ and $\boldsymbol{\rho}_t$ introduced into the iterates $\mathbf{P}_{t+1}$ and $\mathbf{Q}_{t+1}$, respectively. Note that $\mathbf{A}_\phi$ and $\mathbf{A}_\psi$ are asymmetric. We then can leverage the Schur decompositions of the matrix pairs in non-Euclidean metrics $\mathbf{B}_\phi$ or $\mathbf{B}_\psi$ to get the iteration complexity $T = O(\sqrt{\frac{\sigma_k^2}{\sigma_k^2-\sigma_{k+1}^2}}\log\frac{1}{\epsilon(\sigma_k^2-\sigma_{k+1}^2)\cos\theta_0})$. Besides, the errors $\boldsymbol{\delta}_t$ and $\boldsymbol{\rho}_t$ can be well controlled by the least-squares solver, according to the following two lemmas.

Lemma 3.2 *(Xu and Li, 2019) Consider the least-squares subproblem $\min_{\mathbf{\Phi}} l_t(\mathbf{\Phi})$, for which the minimizer and objective sub-optimality gap can be expressed as $\mathbf{\Phi}_t^\star = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{\Psi}_t$ and $\epsilon_t(\mathbf{\Phi}) = l_t(\mathbf{\Phi}) - l_t(\mathbf{\Phi}_t^\star) = \frac{1}{2}\|\mathbf{\Phi}-\mathbf{\Phi}_t^\star\|_{\mathbf{C}_{xx},F}^2$, respectively. We have that*

$$\epsilon_t(\mathbf{\Phi}_t^{(0)}) \le 2k\sigma_1^2\tan^2\max\{\theta_{\max}(\mathbf{\Phi}_t,\mathbf{U}),\theta_{\max}(\mathbf{\Psi}_t,\mathbf{V})\},$$

for the initial sub-optimality, and accelerated gradient descent takes $O\big(\mathrm{nnz}(\mathbf{Y})+\mathrm{nnz}(\mathbf{X})\kappa^{\frac{1}{2}}(\mathbf{C}_{xx})\log\frac{\epsilon_t(\mathbf{\Phi}_t^{(0)})}{\epsilon_t(\widehat{\mathbf{\Phi}}_t)}\big)$ complexity to get the final sub-optimality $\epsilon_t\big(\widehat{\mathbf{\Phi}}_t\big)$, where $\mathbf{\Phi}_t^{(0)} = \mathbf{\Phi}_t(\mathbf{\Phi}_t^\top\mathbf{C}_{xx}\mathbf{\Phi}_t)^{-1}(\mathbf{\Phi}_t^\top\mathbf{C}_{xy}\mathbf{\Psi}_t)$ and $\|\mathbf{A}\|_{\mathbf{B},F} = \|\mathbf{B}^{\frac{1}{2}}\mathbf{A}\|_F$. Parallel results hold for $\min_{\mathbf{\Psi}} h_t(\mathbf{\Psi})$ as well.

Lemma 3.3 *Consider the least-squares subproblem $\min_{\mathbf{\Phi}}\widehat{l}_t(\mathbf{\Phi})$, for which the minimizer and the objective sub-optimality gap can be expressed as $\widehat{\mathbf{\Phi}}_t^\star = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\widehat{\mathbf{\Psi}}_t$ and $\widehat{\epsilon}_t(\mathbf{\Phi}) = \widehat{l}_t(\mathbf{\Phi}) - \widehat{l}_t(\widehat{\mathbf{\Phi}}_t^\star) = \frac{1}{2}\|\mathbf{\Phi}-\widehat{\mathbf{\Phi}}_t^\star\|_{\mathbf{C}_{xx},F}^2$. We have that*

$$\widehat{\epsilon}_t(\widehat{\mathbf{\Phi}}_t^{(0)}) \le 8k(\sigma_1^2+\|\boldsymbol{\xi}_t\|_{\mathbf{C}_{yy},2}^2)\tan^2\widehat{\theta}_t,$$

Table 1: Comparison of theoretical convergence rates. $\tilde{O}(\cdot)$ hides log factors for brevity.

	CCALin (Ge et al., 2016)	ALS (Xu and Li, 2019)	FALS (Xu and Li, 2019)	accALS (Ours)
# iterations	$\tilde{O}(\frac{1}{\rho})$	$\tilde{O}(\frac{1}{\rho})$	$\tilde{O}(\frac{1}{\rho})$	$\tilde{O}(\frac{1}{\sqrt{\rho}})$
runtime (AGD)	$\tilde{O}(\frac{ndk\sqrt{\kappa}}{\rho})$	$\tilde{O}(\frac{ndk\sqrt{\kappa}}{\rho})$	$\tilde{O}(\frac{ndk\sqrt{\kappa}}{\rho})$	$\tilde{O}(\frac{ndk\sqrt{\kappa}}{\sqrt{\rho}})$
runtime (SVRG)	$\tilde{O}(\frac{dk(n+\bar{\kappa})}{\rho})$	$\tilde{O}(\frac{dk(n+\bar{\kappa})}{\rho})$	$\tilde{O}(\frac{dk(n+\bar{\kappa})}{\rho})$	$\tilde{O}(\frac{dk(n+\bar{\kappa})}{\sqrt{\rho}})$
runtime (accSVRG)	$\tilde{O}(\frac{dk(n+\sqrt{n\bar{\kappa}})}{\rho})$	$\tilde{O}(\frac{dk(n+\sqrt{n\bar{\kappa}})}{\rho})$	$\tilde{O}(\frac{dk(n+\sqrt{n\bar{\kappa}})}{\rho})$	$\tilde{O}(\frac{dk(n+\sqrt{n\bar{\kappa}})}{\sqrt{\rho}})$
global convergence	yes	yes	no	yes

where $\rho = \frac{\sigma_k - \sigma_{k+1}}{\sigma_k} \in (0,1)$, $\kappa = \max\{\kappa(\mathbf{C}_{xx}), \kappa(\mathbf{C}_{yy})\}$, and $\bar{\kappa} = \max\{\frac{\max_i \|\mathbf{x}_i\|_2^2}{\lambda_{\min}(\mathbf{C}_{xx})}, \frac{\max_i \|\mathbf{y}_i\|_2^2}{\lambda_{\min}(\mathbf{C}_{yy})}\}$.

for the initial sub-optimality, and accelerated gradient descent takes $O\big(\mathrm{nnz}(\mathbf{Y}) + \mathrm{nnz}(\mathbf{X})\kappa^{\frac{1}{2}}(\mathbf{C}_{xx}) \log \frac{\widehat{\epsilon}_t(\widehat{\mathbf{\Phi}}_t^{(0)})}{\widehat{\epsilon}_t(\widehat{\mathbf{\Phi}}_t)}\big)$ *complexity to get the final sub-optimality* $\widehat{\epsilon}_t(\widehat{\widehat{\mathbf{\Phi}}}_t)$*, where* $\widehat{\mathbf{\Phi}}_t^{(0)} = \widehat{\mathbf{\Phi}}_t(\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xx} \widehat{\mathbf{\Phi}}_t)^{-1}(\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xy} \widehat{\mathbf{\Psi}}_t)$*,* $\|\mathbf{A}\|_{\mathbf{B},2} = \|\mathbf{B}^{\frac{1}{2}}\mathbf{A}\|_2$*, and* $\widehat{\theta}_t = \max\{\theta_{\max}(\widehat{\mathbf{\Phi}}_t, \mathbf{U}), \theta_{\max}(\widehat{\mathbf{\Psi}}_t, \mathbf{V})\}$*. Parallel results hold for* $\min_{\mathbf{\Psi}} \widehat{h}_t(\mathbf{\Psi})$ *as well.*

Note that Lemmas 3.2-3.3 handle the errors with the plain alternating least-squares updates and the faster alternating least-square updates, respectively, in Algorithm 1. Particularly, Lemma 3.3 shows that the error $\boldsymbol{\xi}_t$ contained in $\widehat{\mathbf{\Psi}}_t$ from the plan alternating least-squares update will be brought to the faster alternating least-square update via the warm start. The iteration complexity and the complexities of least-squares solvers in a single iteration jointly give us the overall running time of Algorithm 1. Last, when the iterates $\mathbf{P}_T$ and $\mathbf{Q}_T$ get sufficiently close to $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$, respectively, we can conclude that $\mathbf{\Phi}_T$ and $\mathbf{\Psi}_T$ also have been sufficiently close to the target $\mathbf{U}$ and $\mathbf{V}$, respectively, by the following lemma.

Lemma 3.4 *If* $\sin\max\{\theta_{\max}(\mathbf{P}_T, \widetilde{\mathbf{U}}), \theta_{\max}(\mathbf{Q}_T, \widetilde{\mathbf{V}})\} < \frac{\sigma_k^+ \epsilon}{k\sqrt{1+(\sigma_k^+)^2}}$ *where* $\sigma_k^+ = \frac{\sigma_k^2 + \sqrt{\sigma_k^4 - 4\beta}}{2}$*, it holds that* $\sin\max\{\theta_{\max}(\mathbf{\Phi}_T, \mathbf{U}), \theta_{\max}(\mathbf{\Psi}_T, \mathbf{V})\} < \epsilon$.

□

Remark 1 The complexity of the plain alternating least-squares (ALS) in the same setting (Xu and Li, 2019) is

$$O\Big(\frac{k\sigma_k^2}{\sigma_k^2 - \sigma_{k+1}^2}\mathrm{nnz}(\mathbf{X},\mathbf{Y})\kappa^{\frac{1}{2}}(\mathbf{X},\mathbf{Y})\big(\log\frac{1}{(\sigma_k^2-\sigma_{k+1}^2)\cos\theta_0} \cdot \log\frac{1}{\cos\theta_0} + \log\frac{1}{\epsilon}\log\frac{1}{\sigma_k^2-\sigma_{k+1}^2}\big) + \frac{dk^2\sigma_k^2}{\sigma_k^2-\sigma_{k+1}^2}\log\frac{1}{\epsilon\cos\theta_0}\Big),$$

which is a complexity of type $\tilde{O}(\frac{\sigma_k}{\sigma_k - \sigma_{k+1}})$. It's a similar case for the CCALin (Ge et al., 2016). Moreover, due to the difficulty of analysis, the FALS was only shown to have complexity of the same type as ALS in Xu and Li (2019). Contrastingly, the complexity of the accALS (i.e., Algorithm 1) in this work is subsumed within the type of accelerated rates $\tilde{O}(\sqrt{\frac{\sigma_k}{\sigma_k - \sigma_{k+1}}})$, which matches the empirical performance of the faster alternating least-squares for k-CCA. As the same random initializations to $\mathbf{\Phi}_0$ and $\mathbf{\Psi}_0$ are used as with the alternating least-squares in Xu and Li (2019), the accALS algorithm is globally convergent as well.

Remark 2 For comparison to Ge et al. (2016); Xu and Li (2019) which used accelerated gradient descent (AGD) as the least-squares solver to state the running time, we use AGD here as well. In fact, other least-squares solvers, such as stochastic variance reduced gradient (SVRG) and accelerated SVRG (accSVRG), are applicable as well, and corresponding running times can be similarly figured out. All comparisons are summarized in Table 1.

Remark 3 Although the desired momentum parameter $\beta = \sigma_{k+1}^4/4$ is unknown, it could be estimated dynamically during iterations as mentioned in Section 4, and the overall performance remains sufficiently good as indicated by experimental results in Section 5.

4 Practical Implementation

We now consider practical implementation of our accALS algorithm by introducing two sensible strategies proposed in Xu and Li (2019): coupling update equations and adaptively estimating the momentum parameter. The pseudo code is given in Algorithm 2. The coupling strategy simply means the operation of using the latest versions of relevant variables at all occurrences. As for the second strategy, note that we can write the SVD of $\mathbf{C}_{xy}$ as $\mathbf{C}_{xy} = \mathbf{C}_{xx}\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top + \mathbf{U}_\perp\mathbf{\Sigma}_\perp\mathbf{V}_\perp^\top\right)\mathbf{C}_{yy}$, where $(\mathbf{U}_\perp, \mathbf{\Sigma}_\perp, \mathbf{V}_\perp)$

Algorithm 2 Practical implementation of the accALS

Input: data matrices $(\mathbf{X}, \mathbf{Y})$, block size k, iteration number T.

Output: approximate top-k canonical subspaces $(\mathbf{\Phi}_T, \mathbf{\Psi}_T)$.

Set $\mathbf{\Phi}_{-1} = \mathbf{0}$, $\mathbf{\Psi}_{-1} = \mathbf{0}$, $\mathbf{\Phi}_0 = \widetilde{\mathbf{\Phi}}(\widetilde{\mathbf{\Phi}}^\top \mathbf{C}_{xx} \widetilde{\mathbf{\Phi}})^{-\frac{1}{2}}$, $\mathbf{\Psi}_0 = \widetilde{\mathbf{\Psi}}(\widetilde{\mathbf{\Psi}}^\top \mathbf{C}_{yy} \widetilde{\mathbf{\Psi}})^{-\frac{1}{2}}$, where $\widetilde{\mathbf{\Phi}}$ and $\widetilde{\mathbf{\Psi}}$ are entry-wise i.i.d. standard normal

for $t = 0, 1, \cdots, T-1$ **do**

 # Perform plain alternating least-squares updates

$\widehat{\mathbf{\Phi}}_t \approx \arg\min_{\mathbf{\Phi} \in \mathbb{R}^{d_x \times k}} l_t(\mathbf{\Phi})$ which starts from the initial $\mathbf{\Phi}_t(\mathbf{\Phi}_t^\top \mathbf{C}_{xx} \mathbf{\Phi}_t)^{-1}(\mathbf{\Phi}_t^\top \mathbf{C}_{xy} \mathbf{\Psi}_t)$ to approximately minimize $l_t(\mathbf{\Phi}) = \frac{1}{2n}\|\mathbf{X}^\top \mathbf{\Phi} - \mathbf{Y}^\top \mathbf{\Psi}_t\|_F^2 + \frac{r_x}{2}\|\mathbf{\Phi}\|_F^2$

$\widehat{\mathbf{\Psi}}_t \approx \arg\min_{\mathbf{\Psi} \in \mathbb{R}^{d_y \times k}} h_t(\mathbf{\Psi})$ which starts from the initial $\mathbf{\Psi}_t(\mathbf{\Psi}_t^\top \mathbf{C}_{yy} \mathbf{\Psi}_t)^{-1}(\mathbf{\Psi}_t^\top \mathbf{C}_{xy}^\top \widehat{\mathbf{\Phi}}_t)$ to approximately minimize $h_t(\mathbf{\Psi}) = \frac{1}{2n}\|\mathbf{Y}^\top \mathbf{\Psi} - \mathbf{X}^\top \widehat{\mathbf{\Phi}}_t\|_F^2 + \frac{r_y}{2}\|\mathbf{\Psi}\|_F^2$

 # Perform faster alternating least-squares updates

$\beta_{\phi_t} = \frac{1}{4} \min_{1 \le j \le k} (\mathbf{\Sigma}_{jj}^{\phi_t})^2$, where $\mathbf{\Sigma}^{\phi_t} = (\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xx} \widehat{\mathbf{\Phi}}_t)^{-1}(\widehat{\mathbf{\Phi}}_t^\top \mathbf{C}_{xy} \widehat{\mathbf{\Psi}}_t)$

$\widehat{\widehat{\mathbf{\Phi}}}_t \approx \arg\min_{\mathbf{\Phi} \in \mathbb{R}^{d_x \times k}} \widehat{l}_t(\mathbf{\Phi})$ which starts from the initial $\widehat{\mathbf{\Phi}}_t \mathbf{\Sigma}^{\phi_t}$ to approximately minimize $\widehat{l}_t(\mathbf{\Phi}) = \frac{1}{2n}\|\mathbf{X}^\top \mathbf{\Phi} - \mathbf{Y}^\top \widehat{\mathbf{\Psi}}_t\|_F^2 + \frac{r_x}{2}\|\mathbf{\Phi}\|_F^2$

$\mathbf{C}_{xx}$-orthonormalize $(\widehat{\widehat{\mathbf{\Phi}}}_t - \beta_{\phi_t} \mathbf{\Phi}_{t-1})$ such that $(\widehat{\widehat{\mathbf{\Phi}}}_t - \beta_{\phi_t} \mathbf{\Phi}_{t-1}) = \mathbf{\Phi}_{t+1} \mathbf{R}_{t+1}$ where $\mathbf{R}_{t+1} = ((\widehat{\widehat{\mathbf{\Phi}}}_t - \beta_{\phi_t} \mathbf{\Phi}_{t-1})^\top \mathbf{C}_{xx} (\widehat{\widehat{\mathbf{\Phi}}}_t - \beta_{\phi_t} \mathbf{\Phi}_{t-1}))^{1/2}$

$\beta_{\psi_t} = \frac{1}{4} \min_{1 \le j \le k} (\mathbf{\Sigma}_{jj}^{\psi_t})^2$, where $\mathbf{\Sigma}^{\psi_t} = (\widehat{\mathbf{\Psi}}_t^\top \mathbf{C}_{yy} \widehat{\mathbf{\Psi}}_t)^{-1}(\widehat{\mathbf{\Psi}}_t^\top \mathbf{C}_{xy}^\top \widehat{\widehat{\mathbf{\Phi}}}_t)$

$\widehat{\widehat{\mathbf{\Psi}}}_t \approx \arg\min_{\mathbf{\Psi} \in \mathbb{R}^{d_y \times k}} \widehat{h}_t(\mathbf{\Psi})$ which starts from the initial $\widehat{\mathbf{\Psi}}_t \mathbf{\Sigma}^{\psi_t}$ to approximately minimize $\widehat{h}_t(\mathbf{\Psi}) = \frac{1}{2n}\|\mathbf{Y}^\top \mathbf{\Psi} - \mathbf{X}^\top \widehat{\widehat{\mathbf{\Phi}}}_t\|_F^2 + \frac{r_y}{2}\|\mathbf{\Psi}\|_F^2$

$\mathbf{C}_{yy}$-orthonormalize $(\widehat{\widehat{\mathbf{\Psi}}}_t - \beta_{\psi_t} \mathbf{\Psi}_{t-1})$ such that $(\widehat{\widehat{\mathbf{\Psi}}}_t - \beta_{\psi_t} \mathbf{\Psi}_{t-1}) = \mathbf{\Psi}_{t+1} \mathbf{S}_{t+1}$ where $\mathbf{S}_{t+1} = ((\widehat{\widehat{\mathbf{\Psi}}}_t - \beta_{\psi_t} \mathbf{\Psi}_{t-1})^\top \mathbf{C}_{yy} (\widehat{\widehat{\mathbf{\Psi}}}_t - \beta_{\psi_t} \mathbf{\Psi}_{t-1}))^{1/2}$

end for

consists of the $(\mathrm{rank}(\mathbf{C}_{xy}) - k)$ remaining triples of the left singular vector in metric $\mathbf{C}_{xx}$, singular value, and right singular vector in metric $\mathbf{C}_{yy}$, other than those in $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V})$. Thus, it holds that

$$\begin{aligned} \mathbf{\Sigma} &= (\mathbf{U}^\top \mathbf{C}_{xx} \mathbf{U})^{-1} \mathbf{U}^\top \mathbf{C}_{xy} \mathbf{V} \\ &= (\mathbf{V}^\top \mathbf{C}_{yy} \mathbf{V})^{-1} \mathbf{V}^\top \mathbf{C}_{xy}^\top \mathbf{U}, \end{aligned}$$

which motivates us to estimate $\mathbf{\Sigma}$ dynamically as follows:

$$\begin{aligned} \mathbf{\Sigma}^{\phi_t} &= (\mathbf{\Phi}_t^\top \mathbf{C}_{xx} \mathbf{\Phi}_t)^{-1}(\mathbf{\Phi}_t^\top \mathbf{C}_{xy} \mathbf{\Psi}_t), \\ \mathbf{\Sigma}^{\psi_t} &= (\mathbf{\Psi}_t^\top \mathbf{C}_{yy} \mathbf{\Psi}_t)^{-1}(\mathbf{\Psi}_t^\top \mathbf{C}_{xy}^\top \mathbf{\Phi}_t). \end{aligned}$$

Accordingly, $\hat{\sigma}_k^{\phi_t} = \min_{1 \le j \le k} \mathbf{\Sigma}_{jj}^{\phi_t}$ and $\hat{\sigma}_k^{\psi_t} = \min_{1 \le j \le k} \mathbf{\Sigma}_{jj}^{\psi_t}$ are dynamic estimates of σ_k. Note that it almost always holds that $\max\{\hat{\sigma}_k^{\phi_t}, \hat{\sigma}_k^{\psi_t}\} < \sigma_k$ in a finite number of iterations and meantime $\sigma_{k+1} < \sigma_k$, which suggests that we may use $\hat{\sigma}_k^{\phi_t}$ and $\hat{\sigma}_k^{\psi_t}$ as dynamic estimates of σ_{k+1}. For the momentum parameter, the ideal value $\beta = \frac{1}{4}\sigma_{k+1}^4$ implies estimates $\beta_{\phi_t} = \frac{1}{4}(\hat{\sigma}_k^{\phi_t})^4$ and $\beta_{\psi_t} = \frac{1}{4}(\hat{\sigma}_k^{\psi_t})^4$, which however overly underestimates β because $\max\{\hat{\sigma}_k^{\phi_t}, \hat{\sigma}_k^{\psi_t}\} < \sigma_k \le 1$. For compensation, we use $\beta_{\phi_t} = \frac{1}{4}(\hat{\sigma}_k^{\phi_t})^2$ and $\beta_{\psi_t} = \frac{1}{4}(\hat{\sigma}_k^{\psi_t})^2$ instead, which we found works well in practice. More details can be found in Algorithm 2.

5 Experiments

We conduct experiments on three often used real datasets (Snoek et al., 2006; R. Westbury, 1994; Lecun et al., 1998) for CCA (Ge et al., 2016; Wang et al., 2016; Arora et al., 2017; Xu and Li, 2019) to evaluate the accALS Algorithm 2. The data description is given in Table 2. Fixed regularization parameters $r_x = r_y = 0.1$ are used unless otherwise stated. We compare the accALS to two recent variants of the alternating least-squares for k-CCA, i.e., CCALin (Ge et al., 2016) and the practical version of the FALS (Xu and Li, 2019), where $k > 1$ and the latest FALS

Table 2: Statistics of real data.

Data	Description	d_x	d_y	n
Memdiamill	images and its labels	100	120	30000
JW11	acoustic and articulation	273	112	30000
MNIST	images' left and right halves	392	392	60000

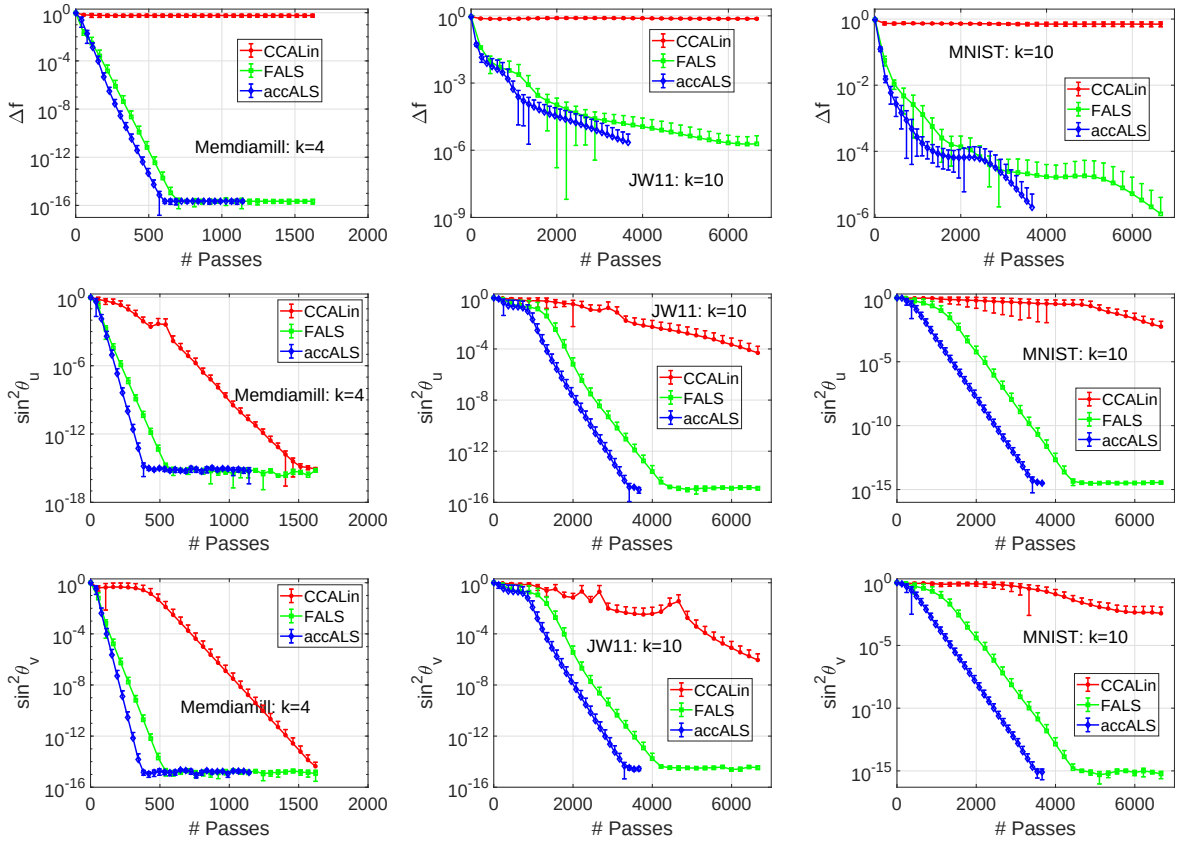

Figure 1: Performance of (faster) alternating least-squares algorithms for k-CCA.

algorithm is well-performed. We use the SVRG as our least-squares solver running 2 epochs for all the CCA algorithms. Each epoch runs n iterations with constant step-sizes $\alpha_\phi = 1/\max_i \|\mathbf{x}_i\|_2^2$ for $\mathbf{\Phi}_t$ and $\alpha_\psi = 1/\max_i \|\mathbf{y}_i\|_2^2$ for $\mathbf{\Psi}_t$, where $\mathbf{x}_i$ represents $\mathbf{X}$'s i-th column. All the algorithms were implemented in MATLAB and run on a laptop. Each k-CCA solver runs $T = 30$ iterations with the same initials which are generated as in Algorithm 2. We use the following three quality measures:

- $\Delta f \triangleq \frac{\mathrm{tr}(\mathbf{\Sigma}) - \mathrm{tr}(\mathbf{\Phi}_t^\top \mathbf{C}_{xy} \mathbf{\Psi}_t)}{\mathrm{tr}(\mathbf{\Sigma})}$, relative error of the objective value in Problem (1),
- $\sin^2\theta_u \triangleq \sin^2\theta_{\max}(\mathbf{\Phi}_t, \mathbf{U})$, squared sine of the largest principal angle between $\mathbf{\Phi}_t$ and $\mathbf{U}$,
- $\sin^2\theta_v \triangleq \sin^2\theta_{\max}(\mathbf{\Psi}_t, \mathbf{V})$, squared sine of the largest principal angle between $\mathbf{\Psi}_t$ and $\mathbf{V}$,

where the ground-truth information $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V})$ is obtained using the output of the MATLAB's svds function for benchmarking purpose. For each measure, smaller is better.

The performance of three algorithms, averaged on 5 runs with different random initializations, is reported in Figure 1 which consists of a 3×3 array of figures with a row for each measure and a column for each dataset. The x-axis is the number of data passes, and the y-axis is the value of a quality measure. The block size used for each dataset is indicated in figures, i.e., $k = 4, 10, 10$ for three datasets, respectively. We can see that faster alternating least-squares algorithms, i.e., FALS and accALS, need significantly less passes over data than the plain alternating least-squares algorithm CCALin. CCALin does not converge in terms of the first measure Δf, because it does not directly solve Problem (1) and thus is not in favour of the first mea-

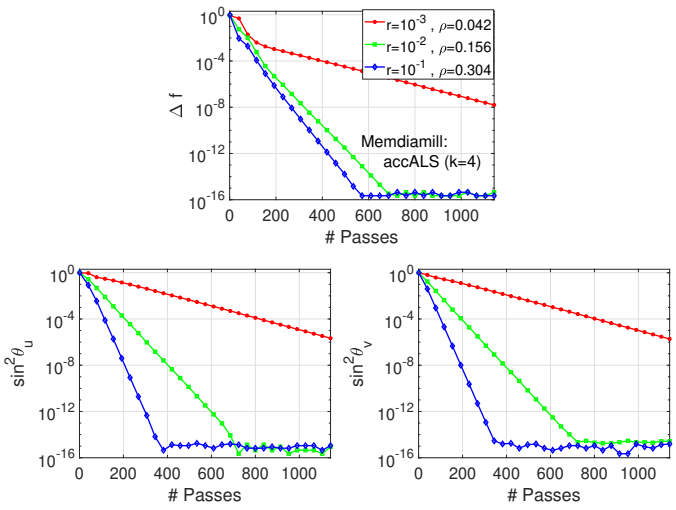

Figure 2: Influence of regularization parameter r on the performance of the accALS.

sure. For the two faster algorithms, according to Table 1, the accALS corresponding to blue curves in figures has performance consistent with the theory (i.e., the nearly square-root dependence of the rate on the relative gap ρ well explains the faster convergence[4]), but the FALS is not the case. Particularly, the accALS is even more pass efficient in terms of each measure across datasets, due to the greater simplicity without loss of the faster convergence rate.

We also would like to study the influence of the regularization parameters r_x and r_y on the performance of the accALS. For simplicity, we fix $r \triangleq r_x = r_y$. Three choices of r, i.e., $r = 10^{-1}, 10^{-2}, 10^{-3}$, are tested on the Memdiamill dataset with $k = 4$. Figure 2 shows the influence of this parameter. It is clear that larger r gives better performance. To understand this phenomenon, we notice that both $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$, and thus also $\mathbf{C} = \mathbf{C}_{xx}^{-1/2}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1/2}$ vary with r. This means that r can affect the relative gap $\rho = \frac{\sigma_k - \sigma_{k+1}}{\sigma_k} \in (0, 1)$ of $\mathbf{C}$. In fact, ρ is increasing with r as shown in the legend of Figure 2, and in turn the performance of the accALS is increasing with ρ by Theorem 3.1, thus demystifying the effect of r.

6 Conclusion

Faster alternating least-squares with momentum acceleration, due to the simplicity and great performance in practice, has been advocated recently for solving the k-CCA problem, though the rationale behind the empirical success has not been well understood yet. To gain a theoretical insight towards fully understanding the rationale, in this work, we examined the two-step update equations of the current faster alternating least-squares and found an even simpler algorithm which only needs to apply momentum at every other iteration on top of the plain alternating least-squares. We demonstrated that the new algorithm has a provably faster linear convergence rate that is characterized by the nearly square-root dependence on the singular value gap of the whitened cross-covariance matrix. Experimental evaluation showed the high consistency between theory and practice of the proposed algorithms, and our algorithm is even more successful than the state-of-the-art in terms of the pass efficiency.

[4]FALS's empirical performance in Figure 1 is much better than that of the CCALin and supposed to have an accelerated rate to match in theory, but its theoretical rate that was achieved previously is only on par with that of the CCALin as shown in Table 1. This gap is filled by our accALS.

Acknowledgements

Authors would like to thank reviewers for their comments that helped improve the quality of the paper.

References

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.

Raman Arora and Teodor Vanislavov Marinov. Efficient convex relaxations for streaming PCA. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10496–10505, Vancouver, Canada, 2019.

Raman Arora, Teodor Vanislavov Marinov, Poorya Mianjy, and Nati Srebro. Stochastic approximation for canonical correlation analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4778–4787, Long Beach, CA, 2017.

Haim Avron, Christos Boutsidis, Sivan Toledo, and Anastasios Zouzias. Efficient dimensionality reduction for canonical correlation analysis. *SIAM J. Scientific Computing*, 36(5), 2014.

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

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT)*, pages 177–186, Paris, France, 2010.

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

Paramveer S. 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.

Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2540–2548, Lille, France, 2015.

Chao Gao, Dan Garber, Nathan Srebro, Jialei Wang, and Weiran Wang. Stochastic canonical correlation analysis. *J. Mach. Learn. Res.*, 20:167:1–167:46, 2019.

Dan Garber, Elad Hazan, Chi Jin, Sham M. Kakade, Cameron Musco, Praneeth Netrapalli, and Aaron Sidford. Faster eigenvector computation via shift-and-invert preconditioning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 2626–2634, New York City, NY, 2016.

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 City, NY, 2016.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, Lake Tahoe, NV, 2013.

Sham M. Kakade and Dean P. Foster. Multi-view regression via canonical correlation analysis. In *Proceedings of 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. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3384–3392, Montreal, Canada, 2015.

Yichao Lu and Dean P. Foster. large scale canonical correlation analysis with iterative least squares. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, Montreal, Canada, 2014.

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.

J R. Westbury. X-ray microbeam speech production database users' handbook. *IEEE Personal Communications - IEEE Pers. Commun.*, 01 1994.

Cees Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM International Conference on Multimedia (MM)*, pages 421–430, Santa Barbara, CA, 2006.

Weiran Wang, Jialei Wang, Dan Garber, and Nati Srebro. Efficient globally convergent stochastic optimization for canonical correlation analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 766–774, Barcelona, Spain, 2016.

Zhiqiang Xu and Ping Li. A practical riemannian algorithm for computing dominant generalized eigenspace. In *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, virtual online.

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

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.