Scalable and Flexible Multiview Canonical Correlation Analysis

Xiao Fu, Kejun Huang, Mingyi Hong, Nicholas D. Sidiropoulos, and Anthony M.-C. So

***Abstract*— This paper considers generalized (multiview) canonical correlation analysis (GCCA) for large-scale datasets. A memory-efficient and computationally lightweight algorithmic framework is proposed for the classic MAX-VAR GCCA formulation as well as its variants. GCCA is gaining renewed interest in various applications such as speech processing and natural language processing. The classic MAX-VAR GCCA problem can be solved optimally via eigen-decomposition of a matrix that compounds the (whitened) correlation matrices of the views. However, this route can easily lead to memory explosion and a heavy computational burden when the size of the views becomes large. In addition, it was unclear how to promote per-specified structure (e.g. sparsity) on the canonical components sought, while structured components analysis is often desired in data analytics. In this work, we propose an alternating optimization (AO)-based algorithm to handle large-scale MAX-VAR GCCA as well as its variations that impose structure on the canonical components. The algorithm avoids instantiating the correlation matrices of the views and thus can achieve substantial saving in memory. It also maintains data sparsity, which can be exploited to alleviate the computational burden. Consequently, the proposed algorithm is highly scalable. Regularization such as sparsity promoting functions can be easily incorporated in the proposed computational framework. Convergence properties of the proposed algorithm are carefully studied. Simulations and large-scale word embedding tasks are employed to showcase the effectiveness of the proposed algorithm.**

I. Introduction

Canonical Correlation Analysis (CCA) [1] produces low dimensional representations by finding common structure of two or more views corresponding to the same entities. A view contains high-dimensional representations of the entities in some domain – e.g., the text and audio representations corresponding to a given word can be considered as different views of this word. CCA is able to deal with views that have different dimensions, and this flexibility is very useful in data fusion, where one is interested in integrating information gathered from different domains. Multiview analysis finds numerous applications in signal processing and machine learning, such as blind source separation [2], [3], direction-of-arrival estimation [4], wireless channel equalization [5], regression [6], clustering [7], speech modeling and recognition [8], [9], and word embedding [10], to name a few. Classical CCA was derived for the two-view case, but Generalized Canonical Correlation Analysis (GCCA) that aims at handling more than two views has a long history as well [11]. A typical application of GCCA is word embedding in natural language processing, where the vocabularies from different languages can be considered as multiple views of the same terms. Word embedding seeks low-dimensional representations of the terms that are well-aligned with human judgment, and applying GCCA to integrate multiple languages was shown to yield better embedding results relative to single-view analyses such as principle component analysis (PCA) [10].

Computationally, GCCA poses interesting and challenging optimization problems. Unlike the two-view case that admits an algebraically simple solution (via eigen-decomposition), GCCA is in general not easily solvable. Many prior works considered the GCCA problem with different cost functions [11], [12], while the proposed algorithms often can only extract a single canonical component and then find others through a deflation process, which is known to suffer from error propagation. Convergence properties of the GCCA algorithms are also largely under-investigated, since the GCCA formulations often involve non-convex constraints (e.g., manifold constraints) that complicate analysis. CCA and GCCA can also pose serious scalability challenges, since they involve auto- and cross-correlations of different views and a whitening stage [13]. These procedures can easily lead to memory explosion and require a large number of flops for computation. They also destroy the sparsity of the data, which is usually what one relies upon to deal with large-scale problems. In recent years, effort has been spent on solving these scalability issues, but the focus is mostly on the two-view case [13]–[15].

Among all different formulations of GCCA, there is a particular one that admits a conceptually simple solution, the so-called MAX-VAR GCCA [11], [12], [16]. MAX-VAR GCCA was first proposed in [11], and its solution amounts to finding the 'directions' aligned to those exhibiting maximum variance for a matrix aggregated from the (whitened) auto-correlations of the views. It can also be viewed as a problem of enforcing *identical* latent representations of different views as opposed to highly correlated ones, which is the more general goal of (G)CCA. The merit of MAX-VAR GCCA is that it can be solved via eigen-decomposition and finds all the canonical components simultaneously (i.e., no deflation is involved). In practice, MAX-VAR GCCA also demonstrates promising performance in various applications such as word embedding [10] and speech recognition [8]. On the other hand, MAX-VAR GCCA has the same scalability problem as the other GCCA formulations: It involves correlation matrices of

X. Fu, K. Huang and N.D. Sidiropoulos are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN55455, e-mail (xfu,huang663,nikos)@umn.edu. M. Hong is with Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa 50011, (515) 294-4111, Email: mingyi@iastate.edu. Anthony M.-C. So is with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, Email: manchoso@se.cuhk.edu.hk

different views and their inverses, which is prohibitive to even instantiate when the data dimension is large. The work in [10] provided a pragmatic way to circumvent this difficulty: PCA was first applied to each view to reduce the rank of the views, and then MAX-VAR GCCA was applied to the rank-truncated views. Such a procedure significantly reduces the number of parameters for characterizing the views and is feasible in terms of memory. However, truncating the rank of the views is prone to information loss, and thus leads to performance degradation.

Besides the basic (G)CCA formulations, *structured* CCA [17] that seeks canonical components with per-specified structure is often considered in applications. For example, sparse CCA is frequently seen in data analytics, for the purpose of discarding irrelevant features when performing CCA [18]–[20]. In multi-lingual word embedding [10], [14], [21], for example, it is known that there are many irrelevant features, such as stop words. In brain activation analysis, it is also believed that many voxels are irrelevant and should be somehow discarded [22], [23]. Gene analysis is another example [18]–[20]. Ideally, CCA seeks a few highly correlated latent components, and so it should naturally be able to identify and down-weight irrelevant features automatically. In practice, however, this ability is often impaired when correlations cannot be reliably estimated, when one only has access to relatively few and/or very noisy samples, or when there is model mismatch due to bad preprocessing (e.g., registration). In those cases, performing feature selection jointly with (G)CCA is well-motivated. However, introducing structure regularizations on the sought canonical components makes the optimization problem even harder – since many regularization terms like the sparsity-promoting regularizers are non-differentiable.

Contributions In this work, our interest lies in solving the MAX-VAR GCCA problem and its variants with structure-promoting regularizers when the dimensions of the views are large. Instead of truncating the rank of the views as in [10], we keep the data *intact* and devise a scalable algorithmic framework to handle the formulated problem. Specifically, our idea is to deal with problem using a two-block alternating optimization (AO) algorithm. Under the AO framework, the proposed algorithm alternates between a regularized least squares subproblem and an orthogonality-constrained subproblem. The merit of this framework is that correlation matrices of the views never need to be explicitly instantiated, and the inversion procedure is avoided. Consequently, the algorithm consumes significantly less memory compared to that required by the original solution using eigen-decomposition and is also very flexible in incorporating different structure-promoting regularizers.

On the theory side, convergence properties of the algorithm are also carefully studied: We first show that the proposed algorithm *globally* converges to a Karush-Kuhn-Tucker (KKT) point of the formulated problem under a variety of regularizers, even when the subproblems are solved in a grossly inexact manner. We also show that the optimality gap shrinks to at most $\mathcal{O}(1/r)$ after r iterations – i.e., at least a sublinear convergence rate can be guaranteed. In addition, we show that when there is no regularization, the proposed algorithm solves the MAX-VAR problem to *global optimality* with a *linear* convergence rate. When the subproblems are not exactly solved, such a rate still holds with some loss in accuracy. The proposed algorithm is applied to synthetic data and a real large-scale word embedding problem, and promising results are observed.

Notation We use $\boldsymbol{X}$ and $\boldsymbol{x}$ to denote a matrix and a vector, respectively. $\boldsymbol{X}(m,:)$ and $\boldsymbol{X}(:,n)$ denote the mth row and the nth column of $\boldsymbol{X}$, respectively; in particular, $\boldsymbol{X}(:,\ell_1:\ell_2)$ denotes a submatrix of $\boldsymbol{X}$ consisting of the ℓ_1-ℓ_2th columns of $\boldsymbol{X}$ (MATLAB notation). $\|\boldsymbol{X}\|_F$ and $\|\boldsymbol{X}\|_p$ for $p \geq 1$ denote the Frobenius norm and the matrix-induced p-norm, respectively. $\|\boldsymbol{X}\|_{p,1} = \sum_{i=1}^{m} \|\boldsymbol{X}(i,:)\|_p$ for $p \geq 1$ denotes the $\ell_p\ell_1$-mixed norm of $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. The superscripts "$T$", "$\dagger$", and "$-1$" denote the matrix operators of transpose, pseudo-inverse and inverse, respectively. The operator $< \boldsymbol{X}, \boldsymbol{Y} >$ denotes the inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ (i.e., $\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{Y})$).

II. Background

The classic two-view CCA can be expressed as the following optimization problem [1]:

$$\begin{aligned} \min_{\boldsymbol{Q}_1,\boldsymbol{Q}_2} \ & \|\boldsymbol{X}_1\boldsymbol{Q}_1 - \boldsymbol{X}_2\boldsymbol{Q}_2\|_F^2 \\ \text{s.t. } & \boldsymbol{Q}_i^T\left(\boldsymbol{X}_i^T\boldsymbol{X}_i\right)\boldsymbol{Q}_i = \boldsymbol{I}, \quad i = 1,2, \end{aligned} \tag{1}$$

where $\boldsymbol{X}_i \in \mathbb{R}^{L\times M_i}$, with its ℓth row $\boldsymbol{X}_i(\ell,:)$ containing the ith view of the ℓth data point, the columns of $\boldsymbol{Q}_i \in \mathbb{R}^{M_i\times K}$ correspond to the K canonical components of view $\boldsymbol{X}_i$, and $(1/L)\boldsymbol{X}_i^T\boldsymbol{X}_i$ serves as an estimate of the correlation of each view. Note that we are essentially maximizing the trace of the estimated cross-correlations between the views, i.e., $\mathrm{Tr}(\boldsymbol{Q}_2^T\boldsymbol{X}_2^T\boldsymbol{X}_1\boldsymbol{Q}_1)$. Thus, $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ can be considered as two linear operators that reduce the dimensionality of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, respectively, so that the reduced-dimension views, i.e., $\boldsymbol{X}_i\boldsymbol{Q}_i$ for $i = 1,2$, are highly correlated. The constraints serve the purpose of normalization. Problem (1) can be solved exactly via an (generalized) eigen-decomposition, but this simple solution only applies to the two-view case. To analyze the case with more than two views, the so-called generalized CCA (GCCA) is often adopted [11]. Different cost functions of GCCA were proposed [11], [12], [24], [25], and arguably the most natural extension of CCA is the following [1]:

$$\begin{aligned} \min_{\{\boldsymbol{Q}_i\}_{i=1}^I} \ & \sum_{i=1}^{I-1}\sum_{j=i+1}^{I} \|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{X}_j\boldsymbol{Q}_j\|_F^2 \\ \text{s.t. } & \boldsymbol{Q}_i^T\left(\boldsymbol{X}_i^T\boldsymbol{X}_i\right)\boldsymbol{Q}_i = \boldsymbol{I}, \ i = 1,\ldots,I, \end{aligned} \tag{2}$$

where I is the number of views. Unfortunately, Problem (2) does not admit an analytic solution. Another formulation of GCCA is more tractable: Instead of forcing pairwise similarity of the reduced-dimension views, one can seek a common latent representation of different views, i.e., [8], [10]–[12], [16]

$$\min_{\{\boldsymbol{Q}_i\}_{i=1}^I, \boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ \sum_{i=1}^{I} (1/2)\,\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\|_F^2\,, \tag{3}$$

where $\boldsymbol{G} \in \mathbb{R}^{L\times K}$ is the common latent representation of the different views. Conceptually, Problems (3) and (2) share

the same goal, i.e., find highly correlated reduced-dimension views. On the other hand, the solutions may be different since the $\boldsymbol{X}_i\boldsymbol{Q}_i$'s yielded by Problem (3) are not forced to be exactly orthogonal, but only approximately so. The upshot of Problem (3) is that it admits a *conceptually* simple algebraic solution, which, as we will show, has the potential to be scaled up to deal with very large problems. In this work, we will focus on Problem (3).

Problem (3) is referred to as the MAX-VAR formulation of GCCA since the optimal solution amounts to taking principal eigenvectors of a matrix aggregated from the correlation matrices of the views. To explain, let us first assume that $\boldsymbol{X}_i$ has full column rank and marginalize $\boldsymbol{Q}_i$ by letting $\boldsymbol{Q}_i = (\sqrt{L})\boldsymbol{X}_i^\dagger\boldsymbol{G}$, where $\boldsymbol{X}_i^\dagger = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T$. By substituting it back to (3), we see that an optimal solution $\boldsymbol{G}_{\rm opt}$ can be obtained via solving the following:

$$\boldsymbol{G}_{\rm opt} = \arg\max_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ {\rm Tr}\left(\boldsymbol{G}^T\left(\sum_{i=1}^I \boldsymbol{X}_i\boldsymbol{X}_i^\dagger\right)\boldsymbol{G}\right). \quad (4)$$

Let $\boldsymbol{M} = \sum_{i=1}^I \boldsymbol{X}_i\boldsymbol{X}_i^\dagger$. Then, an optimal solution is $\boldsymbol{G}_{\rm opt} = \boldsymbol{U}_M(:, 1:K)$, the first K principal eigenvectors of $\boldsymbol{M}$ [26].

Although Problem (3) admits a seemingly easy solution, implementing it for large-scale data is prohibitive. The first difficulty lies in memory: As mentioned, instantiating $\boldsymbol{M} = \sum_{i=1}^I \boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T$ is not doable when L and M_i's are large. The matrix $\boldsymbol{M}$ is an $L \times L$ matrix. In applications like word embedding, L and M_i are the vocabulary size of a language and the number of features defining the terms, respectively, which can both easily exceed $100,000$. This means that the memory for simply instantiating $\boldsymbol{M}$ or $(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}$ can reach 75GB. In addition, even if the views $\boldsymbol{X}_i$ are sparse, computing $(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}$ will create large dense matrices and make it difficult to exploit sparsity in the subsequent processing. To circumvent these difficulties, Rastogi *et al.* [10] proposed to first apply the singular value decomposition (SVD) to the views, i.e., ${\rm svd}(\boldsymbol{X}_i) = \boldsymbol{U}_i\boldsymbol{\Sigma}_i\boldsymbol{V}_i^T$, and then let

$$\hat{\boldsymbol{X}}_i = \boldsymbol{U}_i(:, 1:P)\boldsymbol{\Sigma}_i(1:P, 1:P)(\boldsymbol{V}_i(:, 1:P))^T \approx \boldsymbol{X}_i,$$

where P is much smaller than M_i and L. This procedure enables one to represent the views with significantly fewer parameters, i.e., $(L+M_i+1)P$ compared to LM_i, and allows the original eigen-decomposition based solution to MAX-VAR GCCA to be applied; see more details in [10]. The drawback, however, is also evident: The procedure essentially truncates the rank of the views significantly (since in practice the views almost always have full column-rank, i.e., ${\rm rank}(\boldsymbol{X}_i) = M_i$), and rank-truncation may lose a lot of information. Therefore, it is much more appealing to deal with the original views directly. Another shortcoming of the above approach is that it is not flexible in incorporating regularizations on $\boldsymbol{Q}_i$, while structured $\boldsymbol{Q}_i$ are often desired in different applications [18]–[20].

In this work, we provide an algorithmic framework that deals with the MAX-VAR GCCA problem and its variants with constraints on $\boldsymbol{Q}_i$. We aim at offering simple solutions that are memory-efficient, admit light per-iteration complexity, and feature good convergence properties under certain mild conditions.

III. Proposed Algorithm

In this work, we consider a scalable and flexible algorithmic framework for handling MAX-VAR GCCA and some variants. Specifically, we consider the following formulation

$$\begin{aligned}\min_{\{\boldsymbol{Q}_i\}_{i=1}^I, \boldsymbol{G}} \ & \sum_{i=1}^I (1/2)\left\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\right\|_F^2 + \sum_{i=1}^I g_i\left(\boldsymbol{Q}_i\right), \\ {\rm s.t.} \ & \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I},\end{aligned} \quad (5)$$

where $g_i(\cdot)$ is a regularizer that imposes a certain structure on $\boldsymbol{Q}_i$. Popular regularizers are $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_F$, $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_{2,1}$ and $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_{1,1}$, where $\mu_i \geq 0$ is a regularization parameter for balancing the least squares fitting term and the regularization term. The first regularizer is commonly used for controlling the energy of the dimension-reducing matrix $\boldsymbol{Q}_i$, which also has an effect of improving the conditioning of the least squares problem. The latter two regularizers are used to select features automatically. To be specific, $g_i(\boldsymbol{Q}_i) = \|\boldsymbol{Q}_i\|_{2,1}$ promotes many rows of $\boldsymbol{Q}_i$ to be zero (or approximately zero), and thus can suppress the impact of the corresponding columns (features) in $\boldsymbol{X}_i$ – which is effectively feature selection. The function $g_i(\boldsymbol{Q}_i) = \|\boldsymbol{Q}_i\|_{1,1}$ also does feature selection, but different canonical components may use different features. In this section, we propose an algorithm that can deal with the regularized and the original version of MAX-VAR GCCA under a unified framework.

A. Proposed Algorithm: Alternating Optimization

To deal with Problem (5), our idea is alternating optimization; i.e., we solve two subproblems w.r.t. $\{\boldsymbol{Q}_i\}$ and $\boldsymbol{G}$, respectively. As will be seen, such a simple strategy will leads to highly scalable algorithms in terms of both memory and computational cost.

To begin with, let us consider the subproblem

$$\min_{\boldsymbol{Q}_i} \ (1/2)\left\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}^{(r)}\right\|_F^2 + g_i(\boldsymbol{Q}_i), \ \forall i, \quad (6)$$

where $\boldsymbol{G}^{(r)}$ denotes the iterate of $\boldsymbol{G}$ after the rth iteration. The above problem is a regularized least squares problem. When $\boldsymbol{X}_i$ is large and sparse, many efficient algorithms can be considered to solve it. For example, the alternating direction method of multipliers (ADMM) [27] is frequently employed to handle Problem (6) in a scalable manner. However, ADMM is a primal-dual method that does not guarantee monotonic decrease of the objective value, which will prove useful in later convergence analysis. Hence, we propose to employ a simple proximal gradient method for handling Problem (6). By proximal gradient, we update $\boldsymbol{Q}_i$ by the following update rule:

$$\boldsymbol{Q}_i^{(r+1)} \leftarrow {\tt prox}_{g_i}\left(\boldsymbol{Q}_i^{(r)} - \alpha_i\nabla_{\boldsymbol{Q}_i} f\left(\boldsymbol{Q}^{(r)}, \boldsymbol{G}_i^{(r)}\right)\right), \quad (7)$$

where we define the proximal operator as

$${\tt prox}_g(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} 1/2\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + g(\boldsymbol{x})$$

and use the notations $\boldsymbol{Q} = [\boldsymbol{Q}_1^T, \ldots, \boldsymbol{Q}_I^T]^T$,

$$f(\boldsymbol{Q}, \boldsymbol{G}^{(r)}) = \sum_{i=1}^{I} \frac{1}{2} \left\| \boldsymbol{X}_i \boldsymbol{Q}_i - \boldsymbol{G}^{(r)} \right\|_F^2$$

and $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}, \boldsymbol{G}_i^{(r)})$ as the partial gradient of the least squares part of the objective function w.r.t. $\boldsymbol{Q}_i$, i.e.,

$$\nabla_{\boldsymbol{Q}_i} f\left(\boldsymbol{Q}, \boldsymbol{G}_i^{(r)}\right) = \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{Q}_i - \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}. \tag{8}$$

For many functions $g_i(\cdot)$, the proximity operator in (7) has closed-form or lightweight solutions [28]. For example, for the regularization of interest such as $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_F$, the solution of Problem (7) is simply

$$\boldsymbol{Q}_i^{(r+1)} \leftarrow (\boldsymbol{X}_i^T \boldsymbol{X}_i + \mu_i \boldsymbol{I}) \boldsymbol{Q}_i^{(r)} - \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}. \tag{9}$$

When $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_{2,1}$, the update rule becomes

$$\boldsymbol{Q}_i^{(r+1)}(m,:) \leftarrow \begin{cases} \boldsymbol{0}, & \|\boldsymbol{H}_i(m,:)\|_2 < \mu_i, \\ \left(1 - \frac{\mu_i}{\|\boldsymbol{H}_i(m,:)\|_2}\right) \|\boldsymbol{H}_i(m,:)\|_2, & \text{o.w.}, \end{cases} \tag{10}$$

where $\boldsymbol{H}_i = \boldsymbol{Q}_i^{(r)} - \alpha_i \nabla_{\boldsymbol{Q}_i} f\left(\boldsymbol{Q}^{(r)}, \boldsymbol{G}_i^{(r)}\right)$. For $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_{1,1}$, the update rule is similar to that in (10), which is known as the *soft-thresholding operator*.

By updating $\boldsymbol{Q}_i$ using the rule in (7) (for one time or several times), we obtain $\boldsymbol{Q}_i^{(r+1)}$. Next, we consider solving the subproblem w.r.t. $\boldsymbol{G}$ when fixing $\{\boldsymbol{Q}_i\}_{i=1}^I$. Now we can drop the regularization term since it does not affect the cost value when $\boldsymbol{Q}_i$ is fixed. Then, we have the following equivalence:

$$\begin{aligned} &\arg \min_{\boldsymbol{G}^T \boldsymbol{G} = \boldsymbol{I}} \ \sum_{i=1}^{I} 1/2 \left\| \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} - \boldsymbol{G} \right\|_F^2 \\ \Leftrightarrow &\arg \max_{\boldsymbol{G}^T \boldsymbol{G} = \boldsymbol{I}} \ \operatorname{Tr}\left(\boldsymbol{G}^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I \right). \end{aligned}$$

Therefore, an optimal solution of $\boldsymbol{G}$ is as follows: Let $\boldsymbol{P} = \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)}$. Then, we have

$$\boldsymbol{G}^{(r+1)} \leftarrow \boldsymbol{U}_P \boldsymbol{V}_P^T,$$

where $\boldsymbol{U}_P \boldsymbol{\Sigma}_P \boldsymbol{V}_P^T = \operatorname{svd}(\boldsymbol{P}, '\text{econ}')$, and $\operatorname{svd}(\cdot, '\text{econ}')$ denotes the economy-size SVD that produces $\boldsymbol{U}_P \in \mathbb{R}^{L \times K}$, $\boldsymbol{\Sigma}_P \in \mathbb{R}^{K \times K}$ and $\boldsymbol{V}^T \in \mathbb{R}^{K \times K}$. The above update is optimal in terms of solving the subproblem. In practice, one may also combine the knowledge of the previous iterate $\boldsymbol{G}^{(r)}$ and let

$$\boldsymbol{P} = \gamma \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I + (1 - \gamma) \boldsymbol{G}^{(r)}, \tag{11}$$

where $\gamma \in (0, 1]$. Such a slight modification of forming $\boldsymbol{P}$ does not increase the complexity and is very easy to implement. However, such a simple combination helps establish nice convergence properties of the algorithm, as we will see in the next section.

The algorithm is summarized in Algorithm 1, which we call the alternating optimization-based MAX-VAR GCCA (`AltMaxVar`). As one can see, the algorithm does not instantiate any large dense matrix during the procedure and thus is highly efficient in terms of memory. Also, the procedure does not destroy sparsity of the data, and thus the computational burden is light when the data is sparse – which is often the case in large-scale learning applications. Detailed complexity analysis will be presented in the next subsection.

Algorithm 1: `AltMaxVar`

input : $\{\boldsymbol{X}_i, \mu_i\}_{i=1}^I$; K; T; $(\{\boldsymbol{Q}_i^{(0)}\}_{i=1}^I, \boldsymbol{G}^{(0)})$.

$r \leftarrow 0$;
repeat
  $t \leftarrow 0$;
  $\boldsymbol{E}_i^{(t)} \leftarrow \boldsymbol{Q}_i^{(r)}$ for $i = 1, \ldots, I$;
  while $t < T$ *and convergence not reached* **do**
    for all i, update

$$\boldsymbol{E}_i^{(t+1)} \leftarrow \texttt{prox}_{g_i}\left(\boldsymbol{E}_i^{(t)} - \alpha_i \nabla f_{\boldsymbol{Q}_i}\left(\boldsymbol{E}_i^{(t)}; \boldsymbol{G}_i^{(r)}\right)\right)$$

    where $\nabla f_{\boldsymbol{Q}_i}(\boldsymbol{E}_i^{(t)}; \boldsymbol{G}_i^{(r)}) = -\boldsymbol{X}_i^T \boldsymbol{G}^{(r)} + \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{E}_i^{(t)}$;
    $t \leftarrow t + 1$;
  end
  $\boldsymbol{Q}_i^{(r+1)} \leftarrow \boldsymbol{E}_i^{(t)}$;
  $\boldsymbol{P} \leftarrow \gamma \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I + (1 - \gamma) \boldsymbol{G}^{(r)}$;
  $\boldsymbol{U}_P \boldsymbol{D}_P \boldsymbol{V}_P^T \leftarrow \operatorname{svd}(\boldsymbol{P}, '\text{econ}')$;
  $\boldsymbol{G}^{(r+1)} \leftarrow \boldsymbol{U}_P \boldsymbol{V}_P^T$;
  $r \leftarrow r + 1$;
until *Some stopping criterion is reached*;

output: $\left\{\boldsymbol{Q}_i^{(r)}\right\}_{i=1}^I$, $\boldsymbol{G}^{(r)}$

B. Computational and Memory Complexities

The update rule in (7) inherits the good features from the proximal gradient (PG) method. First, there is no "heavy computation" if the views $\boldsymbol{X}_i$ for $i = 1, \ldots, I$ are sparse. Specifically, the major computation in the update rule of (7) is computing the partial gradient of the smooth part of the cost function, i.e., $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}_i, \boldsymbol{G}_i^{(r)})$. To this end, $\boldsymbol{X}_i \boldsymbol{Q}_i$ should be calculated first, since if $\boldsymbol{X}_i$ is sparse, this matrix multiplication step has a complexity order of $\mathcal{O}(\text{nnz}(\boldsymbol{X}_i) \cdot K)$ flops, where $\text{nnz}(\cdot)$ counts the number of non-zeros. The next multiplication, i.e., $\boldsymbol{X}_i^T(\boldsymbol{X}_i \boldsymbol{Q}_i)$, has the same complexity order. Similarly, the operation of $\boldsymbol{X}_i^T \boldsymbol{G}$ has the same complexity. For solving the $\boldsymbol{G}$-subproblem, the major operation is the SVD of $\boldsymbol{P}$. This step is also not computationally heavy - what we ask for is an economy-size SVD of a very thin matrix (of size $L \times K$, $L \gg K$). This has a complexity order of $\mathcal{O}(LK^2)$ flops [26], which is light.

In terms of memory, all the terms involved (i.e., $\boldsymbol{Q}_i$, $\boldsymbol{G}_i^{(r)}$, $\boldsymbol{X}_i \boldsymbol{Q}_i$, $\boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{Q}_i$ and $\boldsymbol{X}_i^T \boldsymbol{G}_i^{(r)}$) only require $\mathcal{O}(LK)$ memory or less, but the eigen-decomposition-based solution needs $\mathcal{O}(M_i^2)$ and $\mathcal{O}(L^2)$ memory to store $(\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1}$ and $\boldsymbol{M}$, respectively. Note that K is usually very small compared to L and M_i and can be controlled by the designer.

We should mention that the updates of all the $\boldsymbol{Q}_i$'s can be done in parallel, which allows easy distributed implementation. Suppose that each node in a network stores a view $\boldsymbol{X}_i$ and computes $\boldsymbol{Q}_i$ locally. The amount of information exchange after each update of $\boldsymbol{Q}_i$ with other nodes is small – to be precise, the only thing needs to be exchanged is $\boldsymbol{X}_i \boldsymbol{Q}_i$, which has the size of $L \times K$.

IV. Convergence Properties

In this section, we study convergence properties of Algorithm 1. Note that the algorithm alternates between convex and nonconvex set-constrained subproblems and the subproblems may or may not be solved to optimality. Existing convergence analysis for exact and inexact block coordinate descent such as those in [29]–[32] results do not cover the all the considered cases, and thus the convergence properties are not obvious. For the purpose of discussion, we first define a critical point, or, a KKT point, of Problem (5). A KKT point $(\boldsymbol{G}, \boldsymbol{Q})$ satisfies the following optimality conditions:

$$\begin{cases} \mathbf{0} \in \nabla_{\boldsymbol{Q}}\ f(\boldsymbol{Q}, \boldsymbol{G}) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}) \\ \mathbf{0} = \nabla_{\boldsymbol{G}}\ f(\boldsymbol{Q}, \boldsymbol{G}) + \boldsymbol{G}\boldsymbol{\Lambda}, \\ \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}, \end{cases} \tag{12}$$

where $f(\boldsymbol{Q}, \boldsymbol{G}) = \frac{1}{2}\sum_{i=1}^{I} \|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\|_F^2$, $\boldsymbol{\Lambda}$ is a Lagrangian multiplier, $\boldsymbol{Q} = [\boldsymbol{Q}_1^T, \ldots, \boldsymbol{Q}_I^T]^T$ is a collection of all the $\boldsymbol{Q}_i$'s, and

$$\nabla_{\boldsymbol{Q}}\ f(\boldsymbol{Q}, \boldsymbol{G}) = \begin{bmatrix} \nabla_{\boldsymbol{Q}_1} f(\boldsymbol{Q}, \boldsymbol{G}) \\ \vdots \\ \nabla_{\boldsymbol{Q}_I} f(\boldsymbol{Q}, \boldsymbol{G}) \end{bmatrix},\ \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}) = \begin{bmatrix} \partial_{\boldsymbol{Q}_1} g_1(\boldsymbol{Q}_1) \\ \vdots \\ \partial_{\boldsymbol{Q}_I} g_I(\boldsymbol{Q}_I) \end{bmatrix}. \tag{13}$$

and $\partial_{\boldsymbol{Q}_i} g_i(\boldsymbol{Q}_i)$ denotes a subgradient of the (possibly) non-smooth function $g_i(\boldsymbol{Q}_i)$.

Using the above definition, we first show that

Proposition 1 *Assume that $\alpha_i \leq 1/L_i$ for all i, where $L_i = \lambda_{\max}(\boldsymbol{X}_i^T\boldsymbol{X}_i)$ is the largest eigenvalue of $\boldsymbol{X}_i^T\boldsymbol{X}_i$. Also assume that $g_i(\cdot)$ is a closed convex function and that the proximal operator in Line 6 of Algorithm 1 can be solved to optimality, $T \geq 1$, and $\gamma \in (0, 1]$. Then,*

a) *The objective value of Problem* (3) *is non-increasing. In addition, every limit point of the solution sequence $\{\boldsymbol{G}^{(r)}, \{\boldsymbol{Q}_i^{(r)}\}_{i=1}^I\}_{r=0,1,\ldots}$ is a KKT point of Problem* (3).
b) *(Global Convergence) Further assume that $\boldsymbol{X}_i$ and $\boldsymbol{Q}_i^{(0)}$ for $i = 1, \ldots, I$ are bounded and $\mathrm{rank}(\boldsymbol{X}_i) = M_i$. Then, the whole solution sequence converges to the set $\mathcal{K}$ that consists of all the KKT points, i.e., $\lim_{r\to\infty} d^{(r)}(\mathcal{K}) \to 0$, where $d^{(r)}(\mathcal{K}) = \min_{\boldsymbol{Y}\in\mathcal{K}} \|(\boldsymbol{G}^{(r)}, \{\boldsymbol{Q}_i^{(r)}\}) - \boldsymbol{Y}\|_F$.*

The monotonicity of the cost value sequence is a desirable property for very large-size problems. It ensures that the algorithm makes progress in each iteration, which is especially meaningful when the algorithm starts from a good but sub-optimal initialization, e.g., the solution given by the method in [10]. Proposition 1 (a) also characterizes the limit points of the solution sequence. According to Theorem 1, even only one proximal gradient step is performed in each iteration r, every limit point of the algorithm is a KKT point of Problem (5). As we demonstrated in the proof, Algorithm 1 has a similar flavor of *block successive upper bound minimization* (BSUM) [30] and the *block prox-linear* (BPL) framework [31], [32]. However, BSUM does not cover nonconvex constraints such as $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$, and BPL does not cover the $\gamma = 1$ case where the $\boldsymbol{G}$-subproblem is optimally solved. Therefore, the proofs in [30]–[32] cannot be applied to show convergence of Algorithm 1. The (b) part of Proposition 1 establishes the convergence of the whole solution sequence – which is a much stronger result. The assumptions, on the other hand, are also more restrictive, where the views all have full column rank, which was not assumed in the (a) part.

Proposition 1 assures that the algorithm will approach a KKT point asymptotically. In practice, it is also meaningful to estimate the number of iterations that is needed for the algorithm reaching a neighborhood of a KKT point. To this end, let us define the following potential function:

$$\begin{aligned} Z^{(r,r+1)} = &\sum_{t=0}^{T-1} \left\| \nabla_{\boldsymbol{Q}}\ f\left(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}\right) + \partial_{\boldsymbol{Q}} g\left(\boldsymbol{Q}^{(r,t+1)}\right) \right\|_F^2 \\ &+ \left\| \nabla_{\boldsymbol{G}}\ f\left(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}\right) + \boldsymbol{\Lambda}^{(r+1)}\boldsymbol{G}^{(r+1)} \right\|_F^2. \end{aligned} \tag{14}$$

One can see that $Z^{(r,r+1)}$ is a value that is determined by two consecutive outer iterates of the algorithm. $Z^{(r,r+1)}$ has the following property:

Lemma 1 *$Z^{(r,r+1)} \to 0$ implies that $(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)})$ approaches a KKT point.*

The proof of Lemma 1 is in Appendix B. As a result, we can use the value of $Z^{(r,r+1)}$ to measure how close is the current iterate to a KKT point, thereby estimating the iteration complexity. Following this rationale, we show that

Theorem 1 *(Iteration Complexity) Assume that $\alpha_i < 1/L_i$, $0 < \gamma < 1$ and $T \geq 1$. Let $\delta > 0$ and J be the number of iterations needed for that $Z^{(r,r+1)} \leq \delta$ holds for the first time. Then, there exists a constant v such that $\delta \leq v/J-1$; that is, the algorithm converges to a KKT point at least sublinearly.*

The proof of Theorem 1 is relegated to Appendix B. The major message revealed by Theorem 1 is that to reduce the optimality gap (measured by the Z-function) between the current iterate and a KKT point to δ by Algorithm 1, the number of iterations is at most $\mathcal{O}(1/\delta)$. We should mention that the proof is based on worst-case analysis, and the speed in practice maybe much faster. One subtle point that is worth mentioning is that the analysis in Theorem B holds when $\gamma < 1$ – it does not apply to the case where the $\boldsymbol{G}$-subproblem is optimally solved. This reflects some interesting facts in alternating optimization – when subproblems are handled in a more conservative way and using a controlled step size, convergence properties are in general nicer.

Proposition 1 and Theorem 1 characterize convergence properties of the proposed algorithm with a general regularization term $g_i(\cdot)$. It is also interesting to consider the special cases where $g_i(\cdot) = 0$ and $g_i(\cdot) = \mu_i\|\cdot\|_F^2$ – which correspond to the original MAX-VAR formulation and its "diagonal loaded" version [10]. These two cases are *optimally solvable* via eigen-decomposition. Since we posit the problem as a non-convex optimization problem in (3) and use alternating optimization to tackle it for dealing with large-scale cases, a natural question to ask is whether we have traded optimality for scalability. The answer is negative, and the proof is rooted in classic matrix computation theory. To explain, let us denote

$\boldsymbol{U}_1 = \boldsymbol{U}_M(:, 1:K)$ and $\boldsymbol{U}_2 = \boldsymbol{U}_M(:, K+1:L)$ as the K principal eigenvectors of $\boldsymbol{M}$ and the eigenvectors spanning its orthogonal complement, respectively. Recall that our ultimate goal is to find $\boldsymbol{G}$ that is a basis of the range space of $\boldsymbol{U}_M(:, 1:K)$, denoted by $\mathcal{R}(\boldsymbol{U}_M(:, 1:K)) = \mathcal{R}_K(\boldsymbol{M})$. We show that

Lemma 2 *(Global Optimality) Denote the eigenvalues of $\boldsymbol{M} \in \mathbb{R}^{L\times L}$ by $\lambda_1, \ldots, \lambda_L$ in descending order. Consider $g_i(\cdot) = 0$ and $g_i(\cdot) = \mu_i \|\cdot\|_F^2$. Choose $\gamma = 1$. Under the above settings, assume that each subproblem in* (6) *is solved to optimality,* $\mathrm{rank}(\boldsymbol{X}_i) = M_i$ *, $\lambda_K > \lambda_{K+1}$, and $\mathcal{R}(\boldsymbol{G}^{(0)})$ is not orthogonal to any component in $\mathcal{R}_K(\boldsymbol{M})$, i.e.,*

$$\cos(\theta) = \min_{\boldsymbol{u}\in\mathcal{R}_K(\boldsymbol{M}), \boldsymbol{v}\in\mathcal{R}(\boldsymbol{G}^{(0)})} |\boldsymbol{u}^T\boldsymbol{v}| / (\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2) > 0.$$

Then, Algorithm (1) *solves Problem* (3) *optimally when $r \to \infty$. In addition, $\mathcal{R}(\boldsymbol{G}^{(r)})$ approaches $\mathcal{R}_K(\boldsymbol{M})$ linearly, i.e.,*

$$\mathrm{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right) \le \tan(\theta)\left(\lambda_{K+1}/\lambda_K\right)^r,$$

where $\tan(\theta) = \|(\boldsymbol{U}_2^T\boldsymbol{G}^{(0)})(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)})^{-1}\|_2$ *and* $\mathrm{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right) = \|\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}\|_2$.

The proof of Lemma 2 can be found in the appendices. The insight of the proof is as follows: The inner iterations (w.r.t. t) implicitly construct a term in the form of $\boldsymbol{M}\boldsymbol{G}^{(r)} \in \mathbb{R}^{L\times K}$ at iteration r if the $\boldsymbol{Q}$-subproblem is solved to optimality, while the outer loop (w.r.t. r) amounts to a variation of the *Orthogonal Iteration* (OI) algorithm [26], which is a generalization of the power iteration that can estimate the K leading eigenvectors. As such, the outer loop inherits all the properties of the OI algorithm.

The result in Lemma 2 is clearly encouraging: For the MAX-VAR GCCA problem, we can maintain global optimality using low-complexity iterations without memory explosion. In addition, the outer iterations (indexed by r) converges linearly to a global optimal solution – which is rather favorable in terms of convergence rate. On the other hand, the result therein is hardly practical – it holds only if the $\boldsymbol{Q}$-subproblem is solved to optimality which is usually not easy to do when the problem size is large. A natural question is that, if the $\boldsymbol{Q}$-subproblem is solved inexactly, e.g., using a small number of iterations of T, can we expect similar convergence properties? To address this, we further show that

Theorem 2 *Under the same assumptions as those in Lemma 2 except that each subproblem in* (6) *is solved to an accuracy ϵ, i.e., $\|\boldsymbol{Q}_i^{(r+1)} - \tilde{\boldsymbol{Q}}_i^{(r+1)}\|_2 \le \epsilon$, where $\tilde{\boldsymbol{Q}}_i^{(r+1)} = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i\boldsymbol{G}^{(r)}$. Then, after r iterations, we have*

$$\begin{aligned}&\mathrm{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right)\\ &\le \tan(\theta)\left(\lambda_{K+1}/\lambda_K\right)^r + C\left(\sum_{i=1}^{I}\lambda_{\max}(\boldsymbol{X}_i)\epsilon\right),\end{aligned} \tag{15}$$

where C is a constant.

Theorem 2 makes much sense for inexact alternating optimization: in practice, solving the subproblem in (6) is not an easy task, and one may want to stop early (e.g., using a small T). Theorem 2 ensures that if T is large enough to obtain a "good enough" approximation of the solution of Problem (6), the algorithm still converges *linearly* to the desired solution up to some accuracy loss.

V. Numerical Results

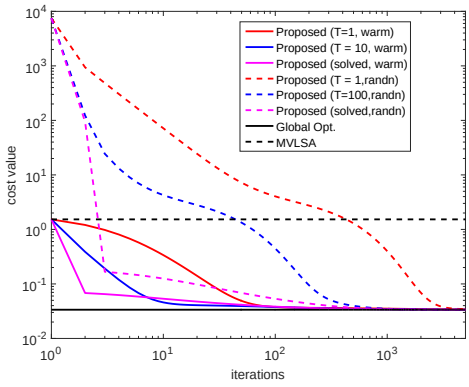
In this section, we use synthetic data and real experiments to showcase the effectiveness of the proposed algorithm. The experiments are carried out using MATLAB on a Linux server with 128GB RAM and 2.0GHz CPU cores.

A. Simulations

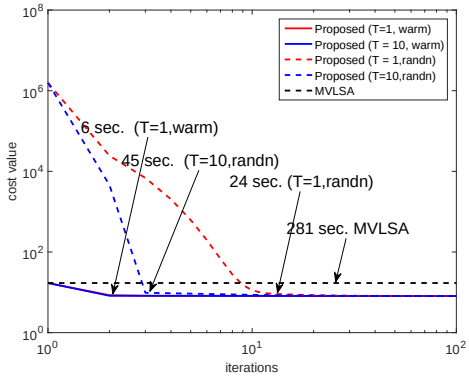
We generate the synthetic data in the following way: First, we let $\boldsymbol{Z} \in \mathbb{R}^{L\times N}$ be a common latent factor of different views, where the entries of $\boldsymbol{Z}$ are drawn from the zero-mean i.i.d. Gaussian distribution and $L \ge N$. Then, a 'mixing matrix' $\boldsymbol{A}_i \in \mathbb{R}^{N\times M}$ is multiplied to $\boldsymbol{Z}$, resulting in $\boldsymbol{Y}_i = \boldsymbol{Z}\boldsymbol{A}_i$. Finally, we add noise so that $\boldsymbol{X}_i = \boldsymbol{Y}_i + \sigma\boldsymbol{N}_i$. Here, $\boldsymbol{A}_i$ and $\boldsymbol{N}_i$ are generated in the same way as $\boldsymbol{Z}$. We apply the algorithm with diagonal loading (i.e., $g_i(\cdot) = \mu_i\|\cdot\|_F^2$) and let $\mu_i = 0.1$. For the synthetic data simulations, we employ the optimal solution that is based on eigen-decomposition and the multiview latent semantic analysis (MVLSA) algorithm that was proposed in [10] as baselines. Recall that MVLSA approximates the views using several leading singular values and vectors.

In Fig. 1(a), we use a small problem instance to examine the convergence properties of the proposed algorithm. Specifically, we let $(L, M, N, I) = (500, 25, 20, 3)$. For such a problem size, the optimal approach via eign-decomposition can be applied. We set $\sigma = 0.1$ in this case, let $P = 8$ for the MVLSA algorithm, and ask for $K = 5$ canonical components. The results are averaged over 50 random trials, where Z, $\{\boldsymbol{A}_i\}$, $\{\boldsymbol{N}_i\}$ are randomly generated in each trial. We test the proposed algorithm under different settings: We let $T = 1$, $T = 10$, and the GD run until the inner loop converges (denoted as 'solved' in the figures). We also initialize the algorithm with random initializations (denoted as 'randn') and warm starts (denoted as 'warm') – i.e., using the solutions of MVLSA as starting points. Some observations from Fig. 1(a) are in order. First, surprisingly, the proposed algorithm using various T's including $T = 1$ and random initialization can reach the global optimum, although Theorem 2 only covers the case where the $\boldsymbol{Q}_i$-subproblem is solved to optimality. Second, by increasing T, the overall cost value decreases faster in terms of number of outer iterations – using $T = 10$ already gives very good speed of decreasing the cost value. Third, MVLSA cannot attain the global optimum, as expected. However, it provides good initialization: Using the warm start, the cost value comes close to the optimal value within 100 iterations in this case, even when $T = 1$ is employed. The cost value does not exhibit linear convergence, since in Theorem 2 the distance $\mathrm{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right)$ is defined by the matrix 2-norm (not Frobenius norm as in the cost function). In the Appendix, we show how $\mathrm{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right)$ evolves along the iterations, and the rate is indeed linear.

Fig. 1(b) shows the cost values against r when applying the algorithms to a relatively large-scale case. Here, we set $L = 100,000, M_i = 50,000, N = 1,000$, and $I = 3$. Each view are randomly generated and with 0.1% non-zero entries. For MVLSA, we let $P = 100$. The task is to seek $K = 10$ canonical components of the views and we stop the proposed algorithm when r reaches 100 and the result is averaged over 10 trials. Note that in this case, the optimal solution cannot be applied because of memory explosion. We see that the proposed algorithm with different settings converges to the same level within a few iterations. Our understanding is that the sparse views are well-conditioned and thus naturally lead to fast convergence of the GD part. We also show the runtimes of the algorithms when reaching the pointed cost levels in Fig. 1(b). Particularly, we see that using $T = 1$ exhibits very competitive runtime performance – Algorithm 1 with $T = 1$ reaches the pointed cost values using 6 seconds and 24 seconds with warm start and random initialization, respectively, while MVLSA uses 281 seconds to yield a worse solution.



(a) Cost v.s. iter. (r)



(b) Cost v.s. iter. (r); large case.

Fig. 1. Performance of the algorithms under various settings.

Fig. 2 shows the dist $\big(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\big)$'s obtained by the proposed algorithm under various settings against r. The settings are the same as those in Fig. 1(a) in the manuscript. We see that if the subproblem w.r.t. $\boldsymbol{Q}_i$ is solved, the algorithm does give a linear convergence rate, as we stated in Theorem 2. Even we restrict the inner loop to $T = 10$ iterations, the rate is still empirically linear, but with a slightly worse slope.

The simulations with outliers go from here

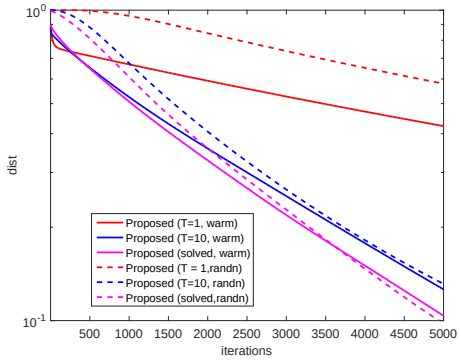


Fig. 2. Linear rate of subspace dist. v.s. iter. (r)

B. Real-Data Validation

We test the algorithms on a large-scale multilingual dataset. The views are extracted from a large word co-occurrence matrix, which is available at https://sites.google.com/a/umn.edu/huang663/research. The original data contains words of three languages, namely, English, Spanish, and French, and all the words are defined by co-occurences with other words. We use the English words to form our first view, $\boldsymbol{X}_1$, which contains $L = 183,034$ words and each word is defined by $M_i = 100,000$ features (co-occurrences). Note that $\boldsymbol{X}_1$ is sparse – only 1.21% of its entries are non-zeros. Using a dictionary, we pick out the translations of the English words contained in $\boldsymbol{X}_1$ in Spanish and French to form $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, respectively. Note that many English words do not have a corresponding word in Spanish (or French). In such cases, we simply let $\boldsymbol{X}_i(\ell,:) = \boldsymbol{0}$ for $i = 2$ (or $i = 3$), resulting in sparser $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$.

Our objective is to find $\boldsymbol{G}$ whose rows are the low-dimensional embeddings of the English words. To evaluate the output, we use the evaluation tool provided at wordvectors.org [33], which runs several word embedding tasks to evaluate a set of given embeddings. Simply speaking, the tasks compare the algorithm-learned embeddings with the judgment of humans and yield high scores if the embeddings are consistent with the humans. The scores are between zero and one, and a score equal to one means a perfect alignment between the learned result and human judgment. We use the result of MVLSA with $P = 640$ as benchmark. The result of applying SVD to $\boldsymbol{X}_1$ without considering different languages is also presented. We apply the proposed algorithm warm started by MVLSA and set $T = 1$. We run two versions of our algorithm. The first one uses $g_i(\cdot) = \|\cdot\|_F^2$ for $i = 1, 2, 3$. The second one uses $g_i(\cdot) = 0.05\|\cdot\|_{2,1}$ for $i = 2, 3$. The reason for adding ℓ_2/ℓ_1 mixed norm regularization to the French and Spanish views is towfold: First, the languages are effectively 'fat matrices' and thus need to use a column-selective regularizer. Second, the ℓ_2/ℓ_1 norm promotes row sparsity of $\boldsymbol{Q}_i$ and thus performs feature selection on $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ – this physically means that we aim at selecting the most useful features from the other languages to help enhance English word embeddings.

Tables I and II show the word embedding results using $K = 50$ and $K = 100$, respectively. We see that using the

information from multiviews does help in improving the word embeddings: For $K = 50$ and $K = 100$, the multiview approaches perform better relative to SVD in 11 and 8 tasks out of 12 tasks. In addition, the proposed algorithm with the regularizer $g_i(\cdot) = \|\cdot\|_F^2$ gives similar or slightly better in average on both experiments compared to MVLSA. The proposed algorithm with the feature-selective regularizer ($g_i(\cdot) = \mu_i\|\cdot\|_{2,1}$) gives the best evaluation results on both experiments – this suggests that for large-scale multiview analysis, feature selection is much meaningful.

TABLE I
EVALUATION ON 12 WORD EMBEDDING TASKS; $K = 50$.

Task	Algorithm ($K = 50$)			
	svd	MVLSA	Proposed ($\|\cdot\|_F^2$)	Proposed ($\|\cdot\|_{2,1}$)
EN-WS-353-SIM	0.63	**0.69**	0.67	0.68
EN-MC-30	0.56	0.63	0.63	**0.64**
EN-MTurk-771	0.54	0.58	0.59	**0.60**
EN-MEN-TR-3k	0.67	0.66	0.67	**0.68**
EN-RG-65	0.51	0.53	0.55	**0.58**
EN-MTurk-287	**0.65**	0.64	**0.65**	0.64
EN-WS-353-REL	0.50	0.51	0.53	**0.55**
EN-VERB-143	0.21	**0.22**	0.21	0.21
EN-YP-130	0.36	0.39	0.38	**0.41**
EN-SIMLEX-999	0.31	**0.42**	0.41	0.39
EN-RW-STANFORD	0.39	**0.43**	**0.43**	**0.43**
EN-WS-353-ALL	0.56	0.59	0.59	**0.60**
average	0.49	0.52	0.53	**0.54**
median	0.53	0.56	0.57	**0.59**

TABLE II
EVALUATION ON 12 WORD EMBEDDING TASKS; $K = 100$.

Task	Algorithm ($K = 100$)			
	svd	MVLSA	Proposed ($\|\cdot\|_F^2$)	Proposed ($\|\cdot\|_{2,1}$)
EN-WS-353-SIM	0.68	**0.72**	0.71	**0.72**
EN-MC-30	0.73	0.68	0.72	**0.74**
EN-MTurk-771	0.59	0.60	0.60	**0.61**
EN-MEN-TR-3k	**0.72**	0.70	0.70	0.71
EN-RG-65	**0.68**	0.63	0.64	**0.68**
EN-MTurk-287	0.61	**0.66**	0.65	0.64
EN-WS-353-REL	**0.57**	0.54	0.55	0.56
EN-VERB-143	0.19	0.28	0.27	**0.29**
EN-YP-130	0.42	0.41	0.41	**0.45**
EN-SIMLEX-999	0.34	**0.42**	0.41	0.41
EN-RW-STANFORD	0.44	**0.46**	0.45	**0.46**
EN-WS-353-ALL	**0.62**	**0.62**	**0.62**	**0.62**
average	0.55	0.56	0.56	**0.58**
median	0.60	0.61	0.61	**0.62**

VI. CONCLUSION

In this work, we revisited the MAX-VAR GCCA problem with an eye towards scenarios involving large-scale and sparse data. The proposed AO-based approach is memory-efficient and has light per-iteration computational complexity if the views are sparse, and is thus suitable for dealing with big data. A thorough convergence analysis was presented, showing that the proposed algorithmic framework guarantees global optimality for the MAX-VAR GCCA problem when the subproblems in the AO framework are exactly solved in each iteration. In addition, when one subproblem is only inexactly solved, global convergence to a KKT point was also shown. Simulations and careful experiments with large-scale multi-lingual data showed that the performance of the proposed algorithm is promising in dealing with large and sparse multiview data.

APPENDIX A
PROOF OF PROPOSITION 1

Before proving the Proposition, let us first simplify the notation. Recall that we have defined $\boldsymbol{Q} = [\boldsymbol{Q}_1^T, \ldots, \boldsymbol{Q}_I^T]^T$ as a collection of $\boldsymbol{Q}_i$'s and we define $F(\boldsymbol{G}, \boldsymbol{Q}) = \sum_{i=1}^I \frac{1}{2}\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\|_F^2 + \sum_{i=1}^I g_i(\boldsymbol{Q}_i)$, and the continuous differentiable part of the above as $f(\boldsymbol{G}, \boldsymbol{Q}) = \sum_{i=1}^I \frac{1}{2}\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\|_F^2$. Since the algorithm is essentially a two-block alternating optimization (i.e., $\boldsymbol{Q}_i$ for all i are updated simultaneously), the above notation suffices to describe the updates. We also define

$$u_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}) + \sum_{i=1}^I < \nabla_{\boldsymbol{Q}_i} f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}), \boldsymbol{Q}_i - \hat{\boldsymbol{Q}}_i > + \sum_{i=1}^I \frac{1}{2\alpha_i}\|\boldsymbol{Q}_i - \hat{\boldsymbol{Q}}\|_F^2 + \sum_{i=1}^I g_i(\boldsymbol{Q}_i);$$

i.e., $u_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right)$ is an approximation of $F(\boldsymbol{G}, \boldsymbol{Q})$ locally at the point $(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}})$. We further define

$$\tilde{u}_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}) + \sum_{i=1}^I < \nabla_{\boldsymbol{Q}_i} f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}), \boldsymbol{Q}_i - \hat{\boldsymbol{Q}}_i > + \sum_{i=1}^I \frac{1}{2\alpha_i}\|\boldsymbol{Q}_i - \hat{\boldsymbol{Q}}\|_F^2;$$

i.e., $\tilde{u}_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right)$ is an approximation of $f(\boldsymbol{G}, \boldsymbol{Q})$ locally at the point $(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}})$. One can see that,

$$\nabla_{\boldsymbol{Q}_i} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = \nabla_{\boldsymbol{Q}_i} \tilde{u}\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right), \tag{16a}$$

$$\nabla_{\boldsymbol{Q}} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) + \partial_{\boldsymbol{Q}} g(\hat{\boldsymbol{Q}}) = \nabla_{\boldsymbol{Q}} \tilde{u}\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) + \partial_{\boldsymbol{Q}} g(\hat{\boldsymbol{Q}}), \tag{16b}$$

where $\nabla_{\boldsymbol{Q}} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right)$, $\partial_{\boldsymbol{Q}} g(\boldsymbol{Q})$ and $\nabla_{\boldsymbol{Q}} \tilde{u}\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right)$ follow the definitions in (13). Since $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{G}, \boldsymbol{Q})$ is L_i-Lipschitz continuous w.r.t. $\boldsymbol{Q}_i$ and $\alpha_i \leq 1/L_i$ for all i, we have the following holds:

$$u_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) \geq F\left(\hat{\boldsymbol{G}}, \boldsymbol{Q}\right), \ \forall\ \boldsymbol{Q}, \tag{17}$$

where the equality holds if and only if $\boldsymbol{Q}_i = \hat{\boldsymbol{Q}}_i$ for all i, i.e.,

$$u_Q\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = F\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right). \tag{18}$$

Now, let us denote by $\boldsymbol{Q}^{(r,t)}$ (where $0 \leq t \leq T$) the solution of $\boldsymbol{Q}$ after t gradient updates when $\boldsymbol{G}^{(r)}$ is fixed, where r is the iteration index of the outer loop. With the above notation, we have $\boldsymbol{Q}^{(r,0)} = \boldsymbol{Q}^{(r)}$ and $\boldsymbol{Q}^{(r,T)} = \boldsymbol{Q}^{(r+1)}$. Also, it can be seen that the update of $\boldsymbol{Q}_i$ can be written as

$$\begin{aligned}\boldsymbol{Q}_i^{(r,t+1)} &= \mathtt{prox}_g\left(\boldsymbol{Q}_i^{(r,t)} - \alpha_i \nabla_{\boldsymbol{Q}_i} f\left(\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r,t)}\right)\right) \\ &= \arg\min_{\boldsymbol{Q}_i}\ u_Q\left(\boldsymbol{Q}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r,t)}\right).\end{aligned} \tag{19}$$

Similarly, we define

$$u_G\left(\boldsymbol{G};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right) = f(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}) + < \nabla_{\boldsymbol{G}} f(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}), \boldsymbol{G}-\hat{\boldsymbol{G}} > + \frac{I}{2\gamma}\left\|\boldsymbol{G}-\hat{\boldsymbol{G}}\right\|_F^2 + \sum_{i=1}^{I} g_i(\hat{\boldsymbol{Q}}_i),$$

where the last term is a constant if $\boldsymbol{Q}$ is fixed. We also have the following holds:

$$\nabla_{\boldsymbol{G}} f\left(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right) = \nabla_{\boldsymbol{G}} u\left(\hat{\boldsymbol{G}};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right) = I\cdot\hat{\boldsymbol{G}} - \sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{Q}_i. \tag{20}$$

The update rule of $\boldsymbol{G}$ in Algorithm 1 can be re-expressed as follows:

$$\begin{aligned}
&\boldsymbol{G}\in\arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \left\|\boldsymbol{G}-\left((1-\gamma)\hat{\boldsymbol{G}}+\gamma\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{Q}_i/I\right)\right\|_F^2\\
\Leftrightarrow\ &\boldsymbol{G}\in\arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \left\|\boldsymbol{G}-\left(\hat{\boldsymbol{G}}-(\gamma/I)\nabla_{\boldsymbol{G}} f\left(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)\right)\right\|_F^2\\
\Leftrightarrow\ &\boldsymbol{G}\in\arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} u_G\left(\boldsymbol{G};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)
\end{aligned}$$

Since $\nabla_{\boldsymbol{G}} f(\boldsymbol{G},\boldsymbol{Q})$ is I-Lipschitz continuous w.r.t. $\boldsymbol{G}$ and $\gamma\le 1$, we have

$$u_G\left(\boldsymbol{G};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)\ge F\left(\boldsymbol{G},\hat{\boldsymbol{Q}}\right),\ \forall\ \boldsymbol{G}, \tag{21}$$

and

$$u_G\left(\hat{\boldsymbol{G}};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right) = F\left(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right). \tag{22}$$

Hence, Algorithm 1 boils down to

$$\boldsymbol{Q}_i^{(r,t+1)} = \arg\min_{\boldsymbol{Q}_i}\ u_Q\left(\boldsymbol{Q};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,t)}\right),\quad t=0,\ldots,T-1 \tag{23a}$$

$$\boldsymbol{G}^{(r+1)}\in\arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}}\ u_G\left(\boldsymbol{G};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,T)}\right). \tag{23b}$$

When $\gamma=1$, (23b) amounts to SVD of $\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{Q}_i/I$ and the $\boldsymbol{G}$-subproblem is optimally solved.

Note that the following holds:

$$\begin{aligned}
F\left(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)}\right) &= u_Q(\boldsymbol{Q}^{(r)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)}) && \text{(24a)}\\
&\ge u_Q(\boldsymbol{Q}^{(r,1)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,0)}) && \text{(24b)}\\
&\ge F\left(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,1)}\right) && \text{(24c)}\\
&= u_Q(\boldsymbol{Q}^{(r,1)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,1)}) && \text{(24d)}\\
&\ge u_Q(\boldsymbol{Q}^{(r,2)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,1)}) && \text{(24e)}\\
&\ \ \vdots && \text{(24f)}\\
&\ge u_Q(\boldsymbol{Q}^{(r+1)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r,T-1)}) && \text{(24g)}\\
&\ge F\left(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r+1)}\right) && \text{(24h)}\\
&= u_G\left(\boldsymbol{G}^{(r)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r+1)}\right) && \text{(24i)}\\
&\ge u_G\left(\boldsymbol{G}^{(r+1)};\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r+1)}\right) && \text{(24j)}\\
&\ge F\left(\boldsymbol{G}^{(r+1)},\boldsymbol{Q}^{(r+1)}\right), && \text{(24k)}
\end{aligned}$$

where (24a) holds because of (18), (24b)-(24g) hold by invoking the update rule (23a) and the properties in (17) and (18), (24h) holds because of (17), (24i) holds due to (22), (24j) is due to the fact that (23b) is optimally solved, and (24k) holds because of (21).

Next, we show that every limit point is a KKT point. Assume that there exists a convergent subsequence of $\{\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)}\}_{r=0,1,\ldots}$, whose limit point is $(\boldsymbol{G}^*,\boldsymbol{Q}^*)$ and the subsequence is indexed by $\{r_j\}_{j=1,\ldots,\infty}$. We have the following chain of inequalities:

$$\begin{aligned}
u_Q\left(\boldsymbol{Q};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j)}\right) &\ge u_Q\left(\boldsymbol{Q}^{(r_j,1)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j)}\right) && \text{(25a)}\\
&\ge u_Q\left(\boldsymbol{Q}^{(r_j,T)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j,T-1)}\right) && \text{(25b)}\\
&\ge F(\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j+1)}) && \text{(25c)}\\
&\ge F\left(\boldsymbol{G}^{(r_j+1)},\boldsymbol{Q}^{(r_j+1)}\right) && \text{(25d)}\\
&\ge F\left(\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})}\right) && \text{(25e)}\\
&= u_Q\left(\boldsymbol{Q}^{(r_{j+1})};\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})}\right), && \text{(25f)}
\end{aligned}$$

where (25a) holds because of the update rule in (23a), (25b) holds by repeating the arguments in (24b)-(24g), (25d) follows (24k), and (25f) is again because of the way that we construct $u_Q(\boldsymbol{Q};\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})})$.

Taking $j\to\infty$, and by continuity of $u_Q(\cdot)$, we have

$$u_Q(\boldsymbol{Q};\boldsymbol{G}^*,\boldsymbol{Q}^*)\ge u_Q(\boldsymbol{Q}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*), \tag{26}$$

i.e., $\boldsymbol{Q}^*$ is a minimum of $u_Q(\boldsymbol{Q};\boldsymbol{G}^*,\boldsymbol{Q}^*)$. Consequently, $\boldsymbol{Q}^*$ satisfies the conditional KKT conditions, i.e.,

$$\boldsymbol{0}\in\nabla_{\boldsymbol{Q}}\tilde{u}_Q(\boldsymbol{Q}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*)+\partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^*),$$

which also means that the following holds:

$$\boldsymbol{0}\in\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{G}^*,\boldsymbol{Q}^*)+\partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^*), \tag{27}$$

following (16).

We now show that $\boldsymbol{Q}^{(r_j,t)}$ for $t=1,\ldots,T$ also converges to $\boldsymbol{Q}^*$. Indeed, we have

$$\begin{aligned}
u_Q(\boldsymbol{Q}^{(r_{j+1})};\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})}) &\le u_Q(\boldsymbol{Q}^{(r_j,1)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j)})\\
&\le u_Q(\boldsymbol{Q}^{(r_j)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j)}),
\end{aligned}$$

where the first inequality was derived from (25). Taking $j\to\infty$, we see that $u_Q(\boldsymbol{Q}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*)\le u_Q(\boldsymbol{Q}^{(r_j,1)};\boldsymbol{G}^*,\boldsymbol{Q}^*)\le u_Q(\boldsymbol{Q}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*)$, which implies that $u_Q(\boldsymbol{Q}^{(r_j,1)};\boldsymbol{G}^*,\boldsymbol{Q}^*)=u_Q(\boldsymbol{Q}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*)\le u_Q(\boldsymbol{Q};\boldsymbol{G}^*,\boldsymbol{Q}^*)$. On the other hand, the problem in (23a) has a unique minimizer when $g_i(\cdot)$ is a convex closed function [28], which means that $\boldsymbol{Q}^{(r_j,1)}\to\boldsymbol{Q}^*$. By recursion, we can show that $\boldsymbol{Q}^{(r_j,t)}$ for $t=1,\ldots,T$ (where T is finite) also converges to $\boldsymbol{Q}^*$ using the same argument. Consequently, we have

$$\boldsymbol{Q}^{(r_j,T)}=\boldsymbol{Q}^{(r_j+1)}\to\boldsymbol{Q}^*. \tag{28}$$

Now, we repeat the proof in (25) to $\boldsymbol{G}$:

$$\begin{aligned} u_G\left(\boldsymbol{G};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j+1)}\right) &\geq u_G\left(\boldsymbol{G}^{(r_j+1)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j+1)}\right) \\ &\geq F(\boldsymbol{G}^{(r_j+1)},\boldsymbol{Q}^{(r_j+1)}) \\ &\geq F\left(\boldsymbol{G}^{(r_j+1)},\boldsymbol{Q}^{(r_j+1)}\right) \\ &= u_G\left(\boldsymbol{G}^{(r_{j+1})};\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})}\right), \end{aligned}$$

Taking $j\to\infty$ and invoking (28), we have

$$u_G\left(\boldsymbol{G};\boldsymbol{G}^*,\boldsymbol{Q}^*\right)\geq u_G\left(\boldsymbol{G}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*\right),\quad \forall \boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}.$$

The above means that $\boldsymbol{G}^*$ satisfies the partial conditional KKT conditions w.r.t. $\boldsymbol{G}$. Combining with (27), we see that $(\boldsymbol{G}^*,\boldsymbol{Q}^*)$ is a KKT point of the original problem.

Now, we show the b) part. First, we show that $\boldsymbol{Q}_i$ remains in a bounded set (the variable $\boldsymbol{G}$ is always bounded since we keep it feasible in each iteration). Since the objective value is non-increasing (cf. Proposition 1), if we denote the initial objective value as V, then $F(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)})\leq V$ holds in all subsequent iterations. Note that when $\boldsymbol{X}_i^{(0)}$ and $\boldsymbol{Q}_i^{(0)}$ are bounded, V is also finite. In particular, we have $\|\boldsymbol{X}_i\boldsymbol{Q}_i-\boldsymbol{G}\|_F^2+2\sum_{i=1}^I g_i(\boldsymbol{Q}_i)\leq 2V$ holds, which implies

$$\|\boldsymbol{X}_i\boldsymbol{Q}_i\|_F\leq\|\boldsymbol{G}\|_F+\sqrt{2V} \tag{30}$$

by the triangle inequality. The right-hand side of (30) is finite since both terms are bounded. Denote $(\|\boldsymbol{G}\|_F+\sqrt{2V})$ by V'. Then, we have

$$\begin{aligned} \|\boldsymbol{Q}_i\|_F &= \|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{X}_i\boldsymbol{Q}_i\|_F \\ &\leq \|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F\cdot\|\boldsymbol{X}_i\boldsymbol{Q}_i\|_F \\ &\leq V'\cdot\|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F. \end{aligned}$$

Now, by the assumption that $\mathrm{rank}(\boldsymbol{X}_i)=M_i$, the term $\|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F$ is bounded. This shows that $\|\boldsymbol{Q}_i\|_F$ is bounded. Hence, starting from a bounded $\boldsymbol{Q}_i^{(0)}$, the solution sequence $\{\{\boldsymbol{Q}_i^{(r)}\},\boldsymbol{G}^{(r)}\}$ remains in a bounded set. Since the constraints of $\boldsymbol{Q}_i$, i.e., $\mathbb{R}^{M_i\times K}$ and $\boldsymbol{G}$ are also closed sets, $\{\{\boldsymbol{Q}_i^{(r)}\},\boldsymbol{G}^{(r)}\}$ remains in a compact set.

Now, let us denote $\mathcal{K}$ as the set containing all the KKT points. Suppose the whole sequence does not converge to $\mathcal{K}$. Then, there exists a convergent subsequence indexed by $\{r_j\}$ such that $\lim_{j\to\infty}d(\mathcal{K})\geq\gamma$ for some positive γ, where $d(\mathcal{K})=\min_{\boldsymbol{Y}\in\mathcal{K}}\|(\boldsymbol{G},\{\boldsymbol{Q}_i\})-\boldsymbol{Y}\|$. Since the subsequence indexed by $\{r_j\}$ lies in a closed and bounded set as we have shown, this subsequence has a limit point. However, as we have shown in Theorem 1, every limit point of the solution sequence is a KKT point. This is a contradiction. Therefore, the whole sequence converges to a KKT point.

Appendix B
Proof of Lemma 1 and Theorem 1

To keep the notation simple, we prove the theorem using the $T=1$ case. The proof of the $T\geq 2$ can be obtained in a straightforward manner. We first show that $Z^{(r,r+1)}\to 0$ implies that a KKT point is reached. First, by the updating rule, we have

$$\begin{aligned} \boldsymbol{Q}^{(r,t)} =& \arg\min_{\boldsymbol{Q}}\ \left\langle\nabla_{\boldsymbol{Q}}f(\boldsymbol{Q}^{(r,t-1)},\boldsymbol{G}^{(r)}),\boldsymbol{Q}-\boldsymbol{Q}^{(r,t-1)}\right\rangle \\ &+\sum_{i=1}^I g_i(\boldsymbol{Q}_i)+\sum_{i=1}^I\frac{1}{2\alpha_i}\|\boldsymbol{Q}_i-\boldsymbol{Q}_i^{(r,t-1)}\|_F^2. \end{aligned} \tag{31}$$

Therefore, there exists a $\partial_{\boldsymbol{Q}}g(\boldsymbol{Q}^{(r,t)})$, $\boldsymbol{Q}^{(r,t)}$ satisfies the following optimality conditions:

$$\boldsymbol{0}=\nabla_{\boldsymbol{Q}}f(\boldsymbol{Q}^{(r,t-1)},\boldsymbol{G}^{(r)})+\partial_{\boldsymbol{Q}}g(\boldsymbol{Q}^{(r,t)})+\boldsymbol{D}(\boldsymbol{Q}^{(r,t)}-\boldsymbol{Q}^{(r,t-1)}),$$

where $\boldsymbol{D}=\mathrm{Diag}\left(\frac{1}{\alpha_1}\boldsymbol{1}_{M_1}^T,\ldots,\frac{1}{\alpha_I}\boldsymbol{1}_{M_I}^T\right)$. Summing over the above through $t=0$ to $t=T-1$ we obtain

$$\begin{aligned} &\sum_{t=0}^{T-1}\left(\nabla_{\boldsymbol{Q}}\ f\left(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)}\right)+\partial_{\boldsymbol{Q}}g\left(\boldsymbol{Q}^{(r,t+1)}\right)\right) \\ &=-\left(\boldsymbol{D}(\boldsymbol{Q}^{(r)}-\boldsymbol{Q}^{(r+1)})\right). \end{aligned} \tag{32}$$

Consequently, we have

$$\begin{aligned} &\sum_{t=0}^{T-1}\left\|\nabla_{\boldsymbol{Q}}\ f\left(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)}\right)+\partial_{\boldsymbol{Q}}g\left(\boldsymbol{Q}^{(r,t+1)}\right)\right\|_F^2\to 0 \\ &\Rightarrow \boldsymbol{Q}_i^{(r,t)}-\boldsymbol{Q}_i^{(r,t+1)}\to\boldsymbol{0},\ \forall\ t=0,\ldots,T-1 \\ &\Rightarrow \boldsymbol{Q}_i^{(r)}-\boldsymbol{Q}_i^{(r+1)}\to 0, \end{aligned}$$

which holds since T is finite. The above means that $\boldsymbol{0}\in\nabla_{\boldsymbol{Q}}f(\boldsymbol{Q}^{(r)},\boldsymbol{G}^{(r)})+\partial_{\boldsymbol{Q}}g(\boldsymbol{Q}^{(r)})$ is satisfied when $Z^{r,r+1}\to 0$.

Recall that the update rule of $\boldsymbol{G}$ is equivalent to solving

$$\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}}\quad\left\|\boldsymbol{G}-\left(\boldsymbol{G}^{(r)}-\gamma/_I\left(\nabla_{\boldsymbol{G}}\ f(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r+1)})\right)\right)\right\|_F^2. \tag{33}$$

Therefore, following the argument in (31), we have

$$\begin{aligned} \boldsymbol{G}^{(r+1)}\in\arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}}\ &<\nabla_{\boldsymbol{G}}f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)}),\boldsymbol{G}-\boldsymbol{G}^{(r)}> \\ &+\frac{1}{2\tilde{\gamma}}\|\boldsymbol{G}-\boldsymbol{G}^{(r)}\|_F^2, \end{aligned} \tag{34}$$

where $\tilde{\gamma}=\gamma/I$ and

$$\begin{aligned} &\nabla_{\boldsymbol{G}}f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)})+\frac{1}{\tilde{\gamma}}\left(\boldsymbol{G}^{(r+1)}-\boldsymbol{G}^{(r)}\right) \\ &+\boldsymbol{G}^{(r+1)}\boldsymbol{\Lambda}^{(r+1)}=\boldsymbol{0} \\ \Rightarrow&\left\|\nabla_{\boldsymbol{G}}f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)})+\boldsymbol{G}^{(r+1)}\boldsymbol{\Lambda}^{(r+1)}\right\|_F^2 \\ &=\frac{1}{\tilde{\gamma}^2}\left\|\boldsymbol{G}^{(r+1)}-\boldsymbol{G}^{(r)}\right\|_F^2. \end{aligned} \tag{35}$$

Combining (35) and (32), we have

$$Z^{(r,r+1)}=\frac{1}{\tilde{\gamma}^2}\left\|\boldsymbol{G}^{(r+1)}-\boldsymbol{G}^{(r)}\right\|_F^2+\sum_{i=1}^I\frac{1}{\alpha_i^2}\left\|\boldsymbol{Q}_i^{(r+1)}-\boldsymbol{Q}_i^{(r)}\right\|_F^2. \tag{36}$$

We see that $Z^{(r,r+1)}\to 0$ implies that $(\boldsymbol{G}^{(r+1)},\boldsymbol{Q}^{(r+1)})\to(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)})$ and that a KKT point is reached.

Second, we show that every iterate of $\boldsymbol{Q}$ and $\boldsymbol{G}$ gives sufficiently large decrease of the overall objective function.

Since $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}, \boldsymbol{G})$ is L_i-Lipschitz continuous for all i, we have the following:

$$\begin{aligned} F(\boldsymbol{Q}^{(r,t+1)}, \boldsymbol{G}^{(r)}) \leq & f(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) \\ & + < \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}), \boldsymbol{Q} - \boldsymbol{Q}^{(r)} > \\ & + \sum_{i=1}^{I} g_i(\boldsymbol{Q}_i) + \sum_{i=1}^{I} \frac{L_i}{2} \left\| \boldsymbol{Q}_i - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \end{aligned} \tag{37}$$

Since $\boldsymbol{Q}^{(r,t)}$ is a minimizer of Problem (31), we also have

$$\begin{aligned} & < \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}), \boldsymbol{Q}^{(r,t+1)} - \boldsymbol{Q}^{(r,t)} > + \sum_{i=1}^{I} g_i(\boldsymbol{Q}_i^{(r+1)}) \\ & + \sum_{i=1}^{I} \frac{1}{2\alpha_i} \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2 \leq \sum_{i=1}^{I} g_i(\boldsymbol{Q}_i^{(r,t+1)}). \end{aligned} \tag{38}$$

Combining (37) and (38), we have

$$\begin{aligned} & F(\boldsymbol{Q}^{(r,t+1)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) \\ & \leq - \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \end{aligned} \tag{39}$$

Summing up the above over $t = 0, \ldots, T-1$,

$$\begin{aligned} & F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}) \\ & \geq \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \end{aligned} \tag{40}$$

By the same derivation, we have

$$\begin{aligned} & F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r+1)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}) \\ & \leq - \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \left\| \boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)} \right\|_F^2, \quad \forall \boldsymbol{G}^T \boldsymbol{G} = \boldsymbol{I}. \end{aligned} \tag{41}$$

Combining (40) and (41), we have

$$\begin{aligned} & F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r+1)}) \\ & \geq \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \left\| \boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)} \right\|_F^2 \\ & + \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \end{aligned} \tag{42}$$

Summing up $F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)})$ over $r = 0, 1, \ldots, J-1$, we have Eq. (43). where $\alpha_{\min} = \min\{\alpha_1, \ldots, \alpha_I\}$ and

$$c = \min \left\{ \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \tilde{\gamma}^2, \left\{ \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \alpha_i^2 \right\}_{i=1,\ldots,I} \right\}.$$

By the definition of J, we have

$$\begin{aligned} & \frac{F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - F(\boldsymbol{Q}^{(J)}, \boldsymbol{G}^{(J)})}{J-1} \geq \frac{\sum_{r=0}^{J-1} c Z^{(r,r+1)}}{J-1} \geq c \cdot \epsilon \\ & \Rightarrow \epsilon \leq \frac{1}{c} \frac{F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - \bar{F}}{J-1} \\ & \Rightarrow \epsilon \leq \frac{v}{J-1}, \end{aligned}$$

where $\bar{F}$ is the lower bound of the cost function and

$$v = \frac{F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - \bar{F}}{c}.$$

This completes the proof.

APPENDIX C
PROOF OF LEMMA 2

If the subproblem w.r.t. $\boldsymbol{Q}_i$ is solved at outer iteration r, then, by the first-order optimality condition and the assumption that $\boldsymbol{X}_i$ has full column rank, we have

$$\boldsymbol{Q}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}. \tag{44}$$

Therefore, the update w.r.t. $\boldsymbol{G}$ is simply

$$\boldsymbol{U}_P^{(r)} \boldsymbol{\Sigma}_P^{(r)} (\boldsymbol{V}_P^{(r)})^T = \text{svd}\left(\boldsymbol{M}\boldsymbol{G}^{(r)}, ' \text{econ}' \right) \tag{45a}$$

$$\boldsymbol{G}^{(r+1)} = \boldsymbol{U}_P^{(r)} (\boldsymbol{V}_P^{(r)})^T. \tag{45b}$$

Since $\boldsymbol{V}_P^{(r)} \in \mathbb{R}^{K \times K}$ is an orthonormal matrix, $\boldsymbol{G}^{(r+1)}$ is an orthogonal basis of $\boldsymbol{M}\boldsymbol{G}^{(r)}$. In other words, there exists an invertible $\boldsymbol{\Theta}^{(r+1)}$ such that

$$\boldsymbol{G}^{(r+1)} \boldsymbol{\Theta}^{(r+1)} = \boldsymbol{M}\boldsymbol{G}^{(r)}. \tag{46}$$

The update rule in (46), is essentially the orthogonal iteration algorithm in [26]. Invoking [26, Theorem 8.2.2], the proof is complete.

APPENDIX D
PROOF OF THEOREM 2

The proof follows the insight of the alternating least squares-based generalized eigen-decomposition in [15] with modifications to accommodate the problem structure of MAX-VAR GCCA. At the rth iteration, ideally, we have $\tilde{\boldsymbol{Q}}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}$ if the inner problem is solved to optimality. In practice, what we have is an inexact solution, i.e., $\boldsymbol{Q}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r)} + \boldsymbol{W}_i^{(r)}$, where we assume that the largest singular value of $\boldsymbol{W}_i^{(r)}$ is bounded by ϵ, i.e., $\|\boldsymbol{W}_i^{(r)}\|_2 \leq \epsilon$. Hence, we see that $\sum_{i=1}^{I} \boldsymbol{X}_i^T \boldsymbol{Q}_i^{(r+1)} = \boldsymbol{M}\boldsymbol{G}^{(r)} + \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}$. By the algorithm under $\gamma = 1$, we have

$$\boldsymbol{G}^{(r+1)} = \left(\boldsymbol{M}\boldsymbol{G}^{(r)} + \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \right) \boldsymbol{\Theta}^{(r)},$$

where $\boldsymbol{\Theta}^{(r)} \in \mathbb{R}^{K \times K}$ is a full-rank matrix since the solution via SVD is a change of bases. Let us denote $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ as orthogonal bases of $\mathcal{R}_K(\boldsymbol{M})$ and its orthogonal complement, respectively. Then, we have

$$\begin{bmatrix} \boldsymbol{U}_1^T \boldsymbol{G}^{(r+1)} \\ \boldsymbol{U}_2^T \boldsymbol{G}^{(r+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \\ \boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_2^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \end{bmatrix} \boldsymbol{\Theta}^{(r)}. \tag{47}$$

Now, we observe the equation on the top of the next page. Note that we can normalize the matrix $\boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}$ as follows

$$\boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} = \sigma \cdot \frac{\boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}}{\|\boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}\|_2} = \sigma \tilde{\boldsymbol{W}}^{(r)}, \tag{49}$$

where σ is bounded by $\sigma \leq \sum_{i=1}^{I} \lambda_{\max}(\boldsymbol{X}_i) \epsilon$ where ϵ denotes the 2-$norm$ upper bound of $\boldsymbol{W}_i^{(r)}$. Using the above notations, we come up with Eq. (50) on the next page, where the equality is obtained by the Taylor expansion.

$$
\begin{aligned}
&F(\boldsymbol{Q}^{(r)},\boldsymbol{G}^{(r)})-F(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r+1)})\\
&\qquad\geq\sum_{r=0}^{J-1}\left(\frac{1}{2\tilde{\gamma}}-\frac{1}{2}\right)\left\|\boldsymbol{G}^{(r+1)}-\boldsymbol{G}^{(r)}\right\|_F^2+\sum_{r=0}^{J-1}\sum_{t=0}^{T-1}\sum_{i=1}^{I}\left(\frac{1}{2\alpha_i}-\frac{L_i}{2}\right)\left\|\boldsymbol{Q}_i^{(r,t+1)}-\boldsymbol{Q}_i^{(r,t)}\right\|_F^2.\\
&\qquad=\sum_{r=0}^{J-1}\left(\frac{1}{2\tilde{\gamma}}-\frac{1}{2}\right)\tilde{\gamma}^2\left\|\nabla_{\boldsymbol{G}}f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)})+\boldsymbol{G}^{(r+1)}\boldsymbol{\Lambda}^{(r+1)}\right\|_F^2\\
&\qquad+\sum_{r=0}^{J-1}\sum_{i=1}^{I}\sum_{t=0}^{T-1}\left(\frac{1}{2\alpha_i}-\frac{L_i}{2}\right)\alpha_i^2\left\|\nabla_{\boldsymbol{Q}_i}f(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)})+\partial_{\boldsymbol{Q}_i}g_i(\boldsymbol{Q}^{(r,t+1)})\right\|_F^2\geq\sum_{r=0}^{J-1}cZ^{(r,r+1)},
\end{aligned}
\tag{43}
$$

$$
\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\right)^{-1}\right\|_2=\left\|\left(\boldsymbol{\Lambda}_2\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}+\boldsymbol{U}_2^T\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{W}_i^{(r)}\right)\left(\boldsymbol{\Lambda}_2\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}+\boldsymbol{U}_2^T\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{W}_i^{(r)}\right)^{-1}\right\|_2. \tag{48}
$$

$$
\begin{aligned}
&\left\|\left(\boldsymbol{\Lambda}_2\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}+\boldsymbol{U}_2^T\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{W}_i^{(r)}\right)\left(\boldsymbol{\Lambda}_1\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}+\boldsymbol{U}_1^T\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{W}_i^{(r)}\right)^{-1}\right\|_2\\
=&\left\|\left(\boldsymbol{\Lambda}_2\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}+\boldsymbol{U}_2^T\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{W}_i^{(r)}\right)\left(\left(\boldsymbol{\Lambda}_1\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}+\sigma\left(\boldsymbol{\Lambda}_1\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\tilde{\boldsymbol{W}}^{(r)}\left(\boldsymbol{\Lambda}_1\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}+\mathcal{O}(\sigma^2)\right)\right\|_2.
\end{aligned}
\tag{50}
$$

$$
\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\right)^{-1}\right\|_2\leq\frac{\lambda_{K+1}}{\lambda_K}\left\|(\boldsymbol{U}_2^T\boldsymbol{G}^{(r)})\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2+\left(\sum_{i=1}^{I}\lambda_{\max}(\boldsymbol{X}_i)\epsilon\right)\mathcal{O}\left(\left\|\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2^2\right). \tag{51}
$$

Now, let us drop the second- and higher-oder terms of σ which are sufficiently small and 'absorb' them in $\mathcal{O}(\|(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)})^{-1}\|_2^2)$. Consequently, we obtain Eq. (51). Now, we show that $\left\|\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2^2$ is bounded for all r. This can be seen by induction. For $r=1$, we see that $\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(1)}\right)^{-1}\right\|_2$ has to be bounded since we assumed that $\operatorname{rank}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)}\right)=K$ and since (51) holds. Using the same argument, we see that for all $r\geq 1$, $\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(1)}\right)^{-1}$ is bounded. Let us denote an upper bound as β, i.e.,

$$
\left\|\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2\leq\beta,\quad\forall r.
$$

Then, we obtain Eq. (52). Using the above, we see Eq. (53) – which completes the proof.

REFERENCES

[1] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[2] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, 2009.

[3] A. Bertrand and M. Moonen, "Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4800–4813, 2015.

[4] Q. Wu and K. M. Wong, "Un-music and un-cle: An application of generalized correlation analysis to the estimation of the direction of arrival of signals in unknown correlated noise," *IEEE Trans. Signal Process.*, vol. 42, no. 9, pp. 2331–2343, 1994.

[5] A. Dogandzic and A. Nehorai, "Finite-length mimo equalization using canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 984–989, 2002.

[6] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *Learning Theory*. Springer, 2007, pp. 82–96.

[7] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 129–136.

[8] R. Arora and K. Livescu, "Multi-view learning with supervision for transformed bottleneck features," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2499–2503.

[9] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 2, pp. 264–277, 2015.

[10] P. Rastogi, B. Van Durme, and R. Arora, "Multiview LSA: Representation learning via generalized cca," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

[11] J. D. Carroll, "Generalization of canonical correlation analysis to three or more sets of variables," in *Proceedings of the 76th annual convention of the American Psychological Association*, vol. 3, 1968, pp. 227–228.

[12] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[13] Z. Ma, Y. Lu, and D. Foster, "Finding linear structure in large datasets with scalable canonical correlation analysis," *arXiv preprint arXiv:1506.08170*, 2015.

[14] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis,"

$$\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\right)^{-1}\right\|_2 \leq \frac{\lambda_{K+1}}{\lambda_K}\left\|(\boldsymbol{U}_2^T\boldsymbol{G}^{(r)})\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2 + \left(\sum_{i=1}^{I}\lambda_{\max}(\boldsymbol{X}_i)\epsilon\right)\mathcal{O}\left(\beta^2\right). \tag{52}$$

$$\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\right)^{-1}\right\|_2 \leq \left(\frac{\lambda_{K+1}}{\lambda_K}\right)^r\left\|(\boldsymbol{U}_2^T\boldsymbol{G}^{(0)})\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)}\right)^{-1}\right\|_2 + \sum_{t=0}^{r-1}\left(\frac{\lambda_{K+1}}{\lambda_K}\right)^t\left(\sum_{i=1}^{I}\lambda_{\max}(\boldsymbol{X}_i)\epsilon\right)\mathcal{O}\left(\beta^2\right). \tag{53}$$

Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 1, pp. 194–200, 2011.

[15] Y. Lu and D. P. Foster, "Large scale canonical correlation analysis with iterative least squares," in *Advances in Neural Information Processing Systems*, 2014, pp. 91–99.

[16] M. Van De Velden and T. H. A. Bijmolt, "Generalized canonical correlation analysis of matrices with missing rows: a simulation study," *Psychometrika*, vol. 71, no. 2, pp. 323–331, 2006.

[17] D. R. Hardoon and J. Shawe-Taylor, "Sparse canonical correlation analysis," *Machine Learning*, vol. 83, no. 3, pp. 331–353, 2011.

[18] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, p. kxp008, 2009.

[19] X. Chen, H. Liu, and J. G. Carbonell, "Structured sparse canonical correlation analysis," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 199–207.

[20] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Statistical applications in genetics and molecular biology*, vol. 8, no. 1, pp. 1–27, 2009.

[21] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation." Association for Computational Linguistics, 2014.

[22] I. Rustandi, M. A. Just, and T. Mitchell, "Integrating multiple-study multiple-subject fmri datasets using canonical correlation analysis," in *Proceedings of the MICCAI 2009 Workshop: Statistical modeling and detection issues in intra-and inter-subject functional MRI data analysis*, 2009.

[23] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

[24] A. Tenenhaus and M. Tenenhaus, "Regularized generalized canonical correlation analysis," *Psychometrika*, vol. 76, no. 2, pp. 257–284, 2011.

[25] A. Tenenhaus, C. Philippe, V. Guillemot, K.-A. Le Cao, J. Grill, and V. Frouin, "Variable selection for generalized canonical correlation analysis," *Biostatistics*, p. kxu001, 2014.

[26] G. H. Golub and C. F. V. Loan., *Matrix Computations*. The Johns Hopkins University Press, 1996.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, pp. 1–122, 2011.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[29] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[30] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[31] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[32] ——, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *arXiv preprint arXiv:1410.1386*, 2014.

[33] M. Faruqui and C. Dyer, "Community evaluation and exchange of word vectors at wordvectors.org," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, USA: Association for Computational Linguistics, June 2014.