SCALABLE AND FLEXIBLE MAX-V AR GENERALIZED CANONICAL CORRELATION ANALYSIS VIA ALTERNATING OPTIMIZATION

Xiao Fu∗, Kejun Huang†, Mingyi Hong∗, Nicholas D. Sidiropoulos∗, Anthony Man-Cho So†

∗Department of ECE, University of Minnesota, Minneapolis, MN, USA
†Department of IME, Iowa State University, Ames, IA, USA

—

ABSTRACT

Unlike dimensionality reduction (DR) tools for single-view data, e.g., principal component analysis (PCA), canonical correlation analysis (CCA) and generalized CCA (GCCA) are able to integrate information from multiple feature spaces of data. This is critical in multi-modal data fusion and analytics, where samples from a single view may not be enough for meaningful DR. In this work, we focus on a popular formulation of GCCA, namely, MAX-V AR GCCA. The classic MAX-V AR problem is optimally solvable via eigen-decomposition, but this solution has serious scalability issues. In addition, how to impose regularizers on the sought canonical components was unclear – while structure-promoting regularizers are often desired in practice. We propose an algorithm that can easily handle datasets whose sample and feature dimensions are both large by exploiting data sparsity. The algorithm is also flexible in incorporating regularizers on the canonical components. Convergence properties of the proposed algorithm are carefully analyzed. Numerical experiments are presented to showcase its effectiveness.

Index Terms— Generalized canonical correlation analysis, MAX-V AR, multi-view analysis

1. INTRODUCTION

In signal processing and data analytics, dimensionality reduction (DR) is usually the first step after signal and data acquisition. Principal component analysis (PCA) is arguably the most popular DR tool. However, PCA is designed to handle data that is acquired from a single feature domain. In modern data science, there are many cases where data have multiple representations in different domains – e.g., a word can be represented as an audio segment, an image, and some video frames. To integrate information from different feature spaces and extract informative low-dimensional representations, canonical correlation analysis (CCA) [1, 2] and generalized canonical correlation analysis (GCCA) [3, 4] are often applied. CCA is widely used in signal processing and data analytics; see [5–10].

Classical CCA considers two views (feature spaces) and formulates the corresponding DR problem as a generalized eigen-decomposition problem. On the other hand, GCCA considers more than two views, and various different formulations exist. Unlike CCA, most GCCA problems (e.g., the sum-of-correlations (SUM-COR) formulation) are NP-hard [11], and so approximations have been proposed to handle them. In the era of Big Data, both GCCA and plain CCA have serious scalability problems, since the computation involves inversion and square-root decomposition of cross-correlation matrices of the views – which is also referred to as the whitening process. Whitening destroys the sparsity of the views, which is often relied upon for dealing with big data, and also creates huge dense matrices that can hardly be stored and greatly increase the computational complexity of subsequent processing.

In recent years, scalability issues of CCA have drawn much attention, but most work focused on the two-view case [12–14]. In this work, we are interested in a popular formulation of GCCA, namely, the MAX-V AR GCCA. Unlike other GCCA formulations, MAX-V AR amounts to computing the leading eigenvectors of an aggregated and whitened correlation matrix of the views – and thus is optimally solvable. MAX-V AR GCCA has gained renewed interest in multilingual word embedding [15] and speech recognition [8], where it has demonstrated promising performance. However, MAX-V AR GCCA has the same scalability issues as the other formulations of (G)CCA. Another challenge of MAX-V AR GCCA is how to incorporate regularizers for promoting presumed or desired structure of the canonical components. Many regularizers are of interest; e.g., sparse canonical components can help discard outlying or irrelevant features, which is useful in gene studies [16–18]. Nonnegativity is of interest in video processing since nonnegative canonical components produce interpretable reduced-dimension data [19, 20].

To address the above challenges, we formulate structure-regularized MAX-V AR GCCA as a non-convex optimization problem and propose an alternating optimization (AO)-based algorithm to handle it. The algorithm alternates between a regularized least squares subproblem and a manifold-constrained non-convex subproblem. This way, the whitening matrices never need to be instantiated and the sparsity of the views is maintained – and thus the algorithm is highly scalable. Analogous to the proximal gradient, the algorithm can handle a variety of structure-promoting regularizers easily. We also carefully study the convergence properties of the proposed algorithm. We show that even when the two subproblems are inexactly solved, the algorithm converges to a critical point globally at a sublinear rate. When the classic MAX-V AR GCCA (without regularization) is considered, we further show that the algorithm in fact approaches a global minimum at a linear rate. Simulations show that the algorithm can easily scale up to views with ~ 100,000 samples and features, which is a substantial improvement from the classic solution that is only suitable for problem sizes of ~ 1, 000.

2. BACKGROUND AND PROBLEM STATEMENT

The classic two-view CCA can be expressed as follows [1]:

\[
\begin{align}
\min_{Q_1, Q_2} & \quad \|X_1 Q_1 - X_2 Q_2\|^2_F \\
\text{s.t.} & \quad Q_i^T (X_i^T X_i) Q_i = I, \quad i = 1, 2,
\end{align}
\]

(1a)

(1b)

where \(X_i \in \mathbb{R}^{L \times M_i}\) represents the \(i\)th view, \(X_i(\ell,:)\) is the high-dimensional data representation of entity (e.g., word) \(\ell\) in view \(i\), \(L\) and \(M_i\) denote the number of entities and the dimension of the \(i\)th
feature space, respectively, \( Q_i \in \mathbb{R}^{M_i \times K} \) contains the canonical components of the \( i \)th view that we aim at finding, and \( K \) is the dimension of the reduced-dimension views. Note that (1) essentially aims at maximizing the correlation of \( X_i Q_1 \) and \( X_i Q_2 \), which is the reason why the problem is called “correlation analysis”. Problem (1) can be solved via the generalized eigen-decomposition, but the reason why the problem is called “correlation analysis”. Problem (2) also aims to find highly correlated reduced-dimension views. Note that (1) essentially like the two-view case, such a pairwise matching criterion has been incorporated in the literature [15].

In this work, we consider a scalable and flexible algorithmic framework for handling MAX-VAR GCCA and its variants with structure-promoting regularizers on \( Q_i \). Specifically, we consider

\[
\min_{\{Q_i\}_i, G^T G = I} \sum_{i=1}^I \| X_i Q_i - G_i \|_F^2 + \sum_{i=1}^I g_i(Q_i),
\]

where \( g_i(\cdot) \) is a regularizer that imposes a certain structure on \( Q_i \).

In the literature, popular regularizers include \( g_i(Q_i) = \mu_i \cdot \| Q_i \|_{2,1} \), \( g_i(Q_i) = \mu_i \cdot \| Q_i \|_2, \) and \( g_i(Q_i) = \mu_i \cdot \| Q_i \|_{2,1} = \mu_i \sum_{m=1}^M \sum_{k=1}^K | Q_{im,k} | \), and \( g_i(Q_i) = 1_1(Q_i) \); i.e., the indicator function of the nonnegative orthant, where \( \mu_i \geq 0 \) is a regularization parameter. We are particularly interested in \( g_i(Q_i) = \mu_i \parallel Q_i \parallel_{2,1} \), since it has the ability of promoting rows of \( Q_i \) to be zero and thus can suppress the impact of the corresponding columns (features) in \( X_i \), which effectively amounts to automatic joint feature selection together with GCCA. 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.

### 3.1. Alternating Optimization

To deal with Problem (4), we build upon an alternating optimization (AO) framework. As we will see, this simple foundation enables us to design highly scalable algorithms in terms of both memory and computational cost, which also features great flexibility in incorporating regularization penalties.

Let us assume that after \( r \) iterations the current iterate is \((Q_i^{(r)}), (G_i^{(r)})\). The subproblem w.r.t. \( Q_i \) is as follows:

\[
\min_{Q_i} \parallel X_i Q_i - G_i^{(r)} \parallel_F^2 + g_i(Q_i), \forall i.
\]

When \( X_i \) is large and sparse, many efficient algorithms can be considered to solve the above – e.g., the alternating direction method of multipliers (ADMM) [25]. However, ADMM does not guarantee monotonic decrease of the objective value if the subproblem is not optimally solved. We wish to maintain monotonicity of the outer loop even when the subproblems are inexact solvability – note that inexact conditional updates are practically unavoidable when dealing with very large problems, for computational complexity considerations. This is an important difference when analyzing big sparse data. Hence, we propose to employ the proximal gradient (PG) method for handling Problem (5). Let us define \( f_i(Q_i, G_i^{(r)}) = \frac{1}{2} \| X_i Q_i - G_i^{(r)} \|_F^2 \) and \( \nabla_{Q_i} f_i(Q_i, G_i^{(r)}) = X_i^T X_i Q_i - X_i^T G_i^{(r)} \). Then, by PG, we update \( Q_i \) by the following update rule:

\[
Q_i^{(r+1)} \leftarrow \text{prox}_{g_i}(Q_i^{(r)} - \alpha_i \nabla_{Q_i} f_i(Q_i^{(r)}, G_i^{(r)})),
\]

where \( \text{prox}_{g_i}(Y) = \arg \min_{X} \| X - Y \|_F^2 + g_i(X) \), and \( Q_i^{(r+1)} \) and \( Q_i^{(r,t)} \) denote \( Q_i \) at iteration \( t + 1 \) and \( t \) when \( G_i^{(r)} \) is fixed, \( t = 0, 1, \ldots, T-1 \), and \( Q_i^{(r,0)} = Q_i^{(r)} \) and \( Q_i^{(r,T)} = Q_i^{(r+1)} \) under this notation. Note that we may choose a small \( T \) for efficiency. For many \( g_i(\cdot) \)‘s including the aforementioned ones, the operator in (6) has closed-form or admits lightweight computation [26].

Next, we consider solving the subproblem w.r.t. \( G \) when fixing \( \{Q_i\}_{i=1}^M \). Instead of dealing with the original \( G \)-subproblem, we propose to solve the following augmented form:

\[
\min_{G^T G = I} \sum_{i=1}^I \frac{1}{2} \| X_i Q_i^{(r+1)} - G_i \|_F^2 + \omega \| G - G_i^{(r)} \|_F^2,
\]

where we define \( \omega = (1-\gamma)/2 \gamma \). Note that when \( \gamma = 1 \), the above boils down to the original \( G \)-subproblem. Adding the proximal term has the effect of controlling step size, which can lead to convergence rate guarantees as will be shown shortly. Expanding the above and dropping the constants, the solution amounts to the
following: Let $R = \gamma \sum_{i=1}^{I} X_i Q(r+1) / \Sigma \gamma + (1 - \gamma) G(r)$. Then, an optimal solution of Problem (7) is

$$G^{(r+1)} \leftarrow U R V_R^T,$$

(8)

where $U_R \Sigma R V_R^T = \text{svd}(R, 'econ')$, and svd(., 'econ') denotes the economy-size SVD that produces $U_R \in \mathbb{R}^{\Sigma \times \Sigma}$, $\Sigma R \in \mathbb{R}^{\Sigma \times K}$, and $V_R \in \mathbb{R}^{K \times K}$. The above solution is based on the well-known Procrustes projection [27].

We call the proposed algorithm in Eqs (6) and (8) alternating optimization-based MAX-VAR GCCA (AltMaxVar). As can be seen, 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.

3.2. Computational and Memory Complexities

If the views $X_i$ for $i = 1, \ldots, I$ are sparse, the PG updates are easy to compute. Specifically, when computing $\nabla Q_i f_i(Q_i, G_i)$, $X_i Q_i$ is calculated first, which has a complexity order of $O(\text{nnz}(X_i) \cdot K)$ flops, where nnz(·) counts the number of non-zeros. The next multiplication, i.e., $X_i^T (X_i Q_i)$, has the same complexity order. The same applies to the operation of $X_i^T G_i$. For solving the $G$-subproblem, since only an economy-size SVD of a very thin matrix (since $L \gg K$) is required, the step only costs $O(LK^2)$ flops [22], which is linear in $L$.

In terms of memory, all the terms involved (i.e., $Q_i$, $G_i$, $X_i \cdot Q_i$, $X_i^T X_i Q_i$, and $X_i^T G_i$) only require $O(LK)$ memory or less, but the eigen-decomposition-based solution needs $O(M_i^2)$ and $O(L^2)$ memory to store $(X_i^T X_i)^{-1}$ and $M_i$, respectively. Note that $K$ is usually very small compared to $L$ and $M_i$.

3.3. Convergence Properties

In this subsection, we present the results of convergence analysis of the proposed AltMaxVar algorithm. Due to space limitations, we will relegate all proofs to the forthcoming journal version 1. Note that the algorithm alternates between a (possibly) non-smooth subproblem and a manifold-constrained subproblem, and the subproblems may or may not be solved to optimality. Existing convergence analyses for exact and inexact block coordinate descent such as those in [28–31] cannot be directly applied to analyze AltMaxVar, and thus these convergence properties are not obvious. We first establish convergence to a Karush-Kuhn-Tucker (KKT) point of Problem (4). A KKT point $(G^*, \{Q_i^*\})$ satisfies the following first-order optimality conditions: $0 \in \nabla Q_i f_i(Q_i^*, G^*) + \partial Q_i g(Q_i^*)$, $\forall i$, and $0 = \sum_{i=1}^{I} \nabla g_i f_i(Q_i^*) + GA^*$, $(G^*)^T G^* = I$, where $A$ is a Lagrangian multiplier associated with the constraint $G^T G = I$, and $\partial Q_i g_i(Q_i^*)$ denotes a set of subgradients of the (possibly) non-smooth function $g_i(Q_i)$. We first show the following:

Proposition 1 Assume that $\alpha_i \leq 1 / L_i$, for all $i$, where $L_i = \lambda_{\max}(X_i^T X_i)$ is the largest eigenvalue of $X_i^T X_i$. Also assume that $g_i(\cdot)$ is a closed convex function, $T \geq 1$, and $\gamma \in (0, 1]$. Then, the following hold: (a) Every limit point of the solution sequence is a KKT point of Problem (2), (b) if $X_i$ and $Q_i^{(0)}$ for $i = 1, \ldots, I$ are bounded and $\text{rank}(X_i) = M$, then the whole solution sequence converges to the set $K$ that consists of all the KKT points.

Proposition 1 (a) characterizes the limit points of the solution sequence: Even if only one proximal gradient step is performed in each iteration $r$, every convergent subsequence of $\{G^{(r)}, \{Q^{(r)}_i\}_i\}$ attains a KKT point of Problem (4). Part (b) shows a stronger result regarding convergence of the whole sequence. The assumptions, on the other hand, are also more restrictive; i.e., $\text{rank}(X_i) = M$.

It is also meaningful to estimate the number of iterations that is needed for the algorithm to reach a neighborhood of a KKT point. To this end, we show the following:

Theorem 1 Assume that $\alpha_i < 1 / L_i$, $0 \leq \gamma < 1$, and $T \geq 1$. Let $\delta > 0$ and $J$ be the number of iterations needed so that $Z^{(r+1)} \leq \delta$ holds for the first time, where

$$Z^{(r+1)} = \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left( \|\nabla Q_i f_i(Q^{(r+1)}_i, G^{(r)}_i)\|_F^2 + \|G^{(r+1)} - G^{(r)}\|_F^2 \right),$$

in which $\nabla Q_i f_i(Q^{(r+1)}_i, G^{(r)}_i) = \frac{1}{\alpha_i} (Q^{(r+1, t)}_i - \text{prox}_{g_i}(Q^{(r, t)}_i - \alpha_i \nabla f_i(Q^{(r, t)}_i, G^{(r)}_i)))$ is the proximal gradient at $Q^{(r+1)}_i$, w.r.t. $Q_i$ and $\alpha_i$ (r+1) is the Lagrangian multiplier associated with $C^{(r+1)}$. Then, there exists a constant $\upsilon > 0$ such that $\delta \leq \upsilon / (\delta - 1)$.

In Theorem 1, the $Z$-function serves as a measure of the optimality gap between the current iterate and a KKT point since one can show that $Z^{(r+1)} \rightarrow 0$ implies that a KKT point is attained. By Theorem 1, AltMaxVar reduces the (measure of the) optimality gap to $O(1/r)$ after $r$ iterations – at least a sublinear rate is guaranteed. One subtle point worth mentioning is that the analysis in Theorem 1 holds when $\gamma < 1$, which corresponds to the case where $\omega > 0$ in the $G$-subproblem and the solution is controlled to be not far away from $G^{(r)}$. This reflects an interesting fact in AO – when the subproblems are handled in a conservative way using a controlled step size, a certain convergence rate property may be guaranteed.

Note that when $g_i(\cdot) = \mu_i \| \cdot \|_F^2$ for $\mu_i > 0$, this case is optimally solvable (i.e., the solution is the $K$ leading eigenvectors of $M = \sum_{i=1}^{I} X_i (X_i^T X_i + \mu_i I)^{-1} X_i^T$ for $\mu_i \geq 0$ [15]). Therefore, it is natural to ask if AltMaxVar loses optimality under such cases by gaining scalability? To address this question, we denote $U_1$ and $U_2$ as the $K$ principal eigenvectors of $M$ and the eigenvectors spanning its orthogonal complement, respectively. Recall that our goal is to find $G$ such that $R(G) = R(U_1)$. We adopt the definition of subspace distance in [22], i.e., dist$(R(G^{(r)}), R(U_1)) = \|U_1 - G^{(r)}\|_2$, where $\|X\|_2$ denotes the matrix 2-norm, and show the following:

Theorem 2 Denote the eigenvalues of $M \in \mathbb{R}^{L \times L}$ by $\lambda_1, \ldots, \lambda_L$ in descending order. Consider $g_i(\cdot) = \mu_i \| \cdot \|_F^2$ for $\mu_i \geq 0$ and let $\gamma = 1$. Assume that $\text{rank}(X_i) = M_i$, $\lambda_K > \lambda_{K+1}$, and $R(G^{(0)})$ is not orthogonal to any component in $R(U_1)$; i.e.,

$$\cos(\theta) = \min_{u \in R(U_1), v \in R(G^{(0)})} \left( \frac{u^T v}{\|u\|_2 \|v\|_2} \right) > 0. \text{ Also assume that each subproblem in (5) is solved to accuracy } \epsilon; \text{ i.e., } \|Q^{(r+1)}_i - Q^{(r+1)}_i\|_2 \leq \epsilon, \text{ where } Q^{(r+1)}_i = (X_i^T X_i + \mu_i I)^{-1} X_i G^{(r)}_i. \text{ Then, after } r \text{ iterations,}
$$

$$\text{dist}(R(G^{(r)}), R(U_1)) \leq \tan(\theta) (\lambda_{K+1} / \lambda_K)^r + C$$

holds, where $C = O \left( \sum_{i=1}^{I} \lambda_{\max}(X_i) \epsilon \right)$ is a constant.

Theorem 2 ensures that if $T$ suffices for the $Q$-subproblem to obtain a good enough approximation to its optimal solution, the algorithm converges linearly to a global optimal solution up to some accuracy loss. In our simulations, we observe that using $T = 1$ already gives very satisfactory results (as will be shown in the next section), which leads to computationally very cheap updates.
4. SIMULATIONS AND CONCLUSIONS

We generate the views by $X_i = Z A_i + \sigma N_i$ where $Z \in \mathbb{R}^{L \times N}$ is common to all views, $A_i \in \mathbb{R}^{N \times M}$ is a “mixing matrix” whose effect is supposed to be suppressed by $Q$, $N_i \in \mathbb{R}^{L \times M}$ is noise, and $\sigma \geq 0$. $Z$, $A_i$, and $N_i$ are large sparse matrices and the non-zero elements follow the zero-mean unit-variance i.i.d. Gaussian distribution. $X_i$ is sparse and its density level $\rho_i$ is defined as $\rho_i = \text{nnz}(X_i)/LM$. In the simulations, we let $\rho = \rho_i$ for all $i$. We use the eigen-decomposition based solution of MAX-VAR GCCA as a benchmark when applicable. Another algorithm called multiview latent semantic analysis (MVLSA) is also employed as a baseline [15].

MVLSA truncates the rank of the views using PCA as pre-processing and then applies the eigen-decomposition based solution.

In Fig. 1, we show the runtime performance of the algorithms for various sizes of the views, where density of the views is controlled so that $\rho \approx 10^{-3}$. The regularization $g(\cdot) = 0.1 \cdot \| \cdot \|^2_F$ is employed by all algorithms. We let $M = L \times 0.8$, $M = N$ and change $M$ from 5,000 to 50,000. To run MVLSA, we truncate the ranks of views to $P = 100$, $P = 500$ and $P = 1,000$, respectively. We use MVLSA with $P = 100$ to initialize AltMaxVar and let $T = 1$ and $\gamma = 1$. We stop the proposed algorithm when the absolute change of the objective value is smaller than $10^{-4}$. Ten random trials are used to obtain the results. One can see that the eigen-decomposition based algorithm does not scale well since the matrix $(X_i^T X_i + \mu I)^{-1}$ is dense. In particular, the algorithm exhausts the memory quota (32GB RAM) when $M = 30,000$. MVLSA with $P = 100$ and the proposed algorithm both scale very well from $M = 5,000$ to $M = 50,000$: When $M = 20,000$, brute-force eigen-decomposition takes almost 80 minutes, whereas MVLSA ($P = 100$) and AltMaxVar both use less than 2 minutes. Note that the runtime of the proposed algorithm already includes the runtime of the initialization time by MVLSA with $P = 100$, and thus the runtime curve of AltMaxVar is slightly higher than that of MVLSA ($P = 100$) in Fig. 1. Another observation is that, although MVLSA exhibits good runtime performance when using $P = 100$, its runtime under $P = 500$ and $P = 1,000$ is not very appealing. The corresponding cost values can be seen in Table 1. The eigen-decomposition based method gives the lowest cost values when applicable, as it is an optimal solution. The proposed algorithm gives favorable cost values that are close to the optimal ones, even when only one iteration of the $Q$-subproblem is implemented for every fixed $G(\cdot)$ - this result supports our analysis in Theorem 2. Increasing $P$ helps improve MVLSA. However, even when $P = 1,000$, the cost value given by MVLSA is still higher than that of AltMaxVar, and MVLSA using $P = 1,000$ is much slower than AltMaxVar.

![Fig. 1: Runtime of the algorithms under various problem sizes.](image)

### Table 1: Cost values of the algorithms under different problem sizes. $L = M/0.8$, $\rho = 10^{-3}$, $\sigma = 0.1$. $\dagger$ means “out of memory.”

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$L=100,000$</th>
<th>$L=500,000$</th>
<th>$L=2,000,000$</th>
<th>$L=5,000,000$</th>
<th>$L=10,000,000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVLSA ($P=100$)</td>
<td>16.843</td>
<td>17.349</td>
<td>16.912</td>
<td>16.785</td>
<td>17.612</td>
</tr>
<tr>
<td>time (min)</td>
<td>0.003</td>
<td>0.009</td>
<td>0.000</td>
<td>0.009</td>
<td>0.003</td>
</tr>
<tr>
<td>Proposed ($\mu=0.05$)</td>
<td>0.478</td>
<td>0.610</td>
<td>0.565</td>
<td>0.775</td>
<td>0.803</td>
</tr>
<tr>
<td>time (min)</td>
<td>3.798</td>
<td>5.425</td>
<td>3.765</td>
<td>24.182</td>
<td></td>
</tr>
<tr>
<td>Proposed ($\mu=1$)</td>
<td>0.942</td>
<td>1.034</td>
<td>0.944</td>
<td>1.265</td>
<td>1.294</td>
</tr>
<tr>
<td>time (min)</td>
<td>2.182</td>
<td>3.714</td>
<td>4.510</td>
<td>16.378</td>
<td></td>
</tr>
<tr>
<td>Proposed ($\mu=5$)</td>
<td>0.003</td>
<td>0.021</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>time (min)</td>
<td>1.735</td>
<td>2.714</td>
<td>3.725</td>
<td>13.447</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 presents the simulation results of a large-scale case in the presence of outlying features. Here, we fix $L = 100,000$ and $M = 80,000$ and change the density level $\rho$. We add 30,000 outlying features to each view and every outlying feature is a random sparse vector whose non-zero elements follow the zero-mean i.i.d. Gaussian distribution. We also scale the outlying features so that the average energy of the clean and outlying features are identical. The other settings follow those in the last simulation. In this case, the optimal solution to Problem (4) is unknown. Therefore, we evaluate the performance by observing metric1 = $\frac{1}{I} \sum_{i=1}^{I} |X_i(:,S_i)| Q_i(S_i,:) - G_i^2_F$, and metric2 = $\frac{1}{I} \sum_{i=1}^{I} |X_i(:,S_i)| Q_i(S_i,:) - G_i^2_F$, where $S_i$ and $S_i$ denote the index sets of clean and outlying features of view $i$, respectively – i.e., $X_i(:,S_i) = Z A_i$ and $X_i(:,S_i) = O$, if noise is absent. metric1 measures the performance of matching $G$ with the relevant part of the views, while metric2 measures the performance of suppressing the irrelevant part. We desire low values of metric1 and metric2 simultaneously. We use $g(\cdot) = \mu_i |\cdot|_2$ for AltMaxVar to discard the outlying features. One can see from Table 2 that the proposed algorithm with $\mu_i = 0.05$ gives the most balanced result – both evaluation metrics are at fairly low levels. Using $\mu_i = 0.5$ suppresses $Q_i(S,:) - G_i$ even better, but using such a relatively large $\mu_i$ degrades the fitting metric. In terms of runtime, one can see that the proposed algorithm operates within the same order of magnitude as MVLSA with $P = 100$. Since AltMaxVar works with the intact views of size $L \times M$ while MVLSA works with significantly reduced-dimension data, such runtime performance of AltMaxVar is very satisfactory.

### Table 2: Evaluation in the presence of outlying features. $L = 100,000$, $M = 80,000$, $|S| = 30,000$, $\sigma = 1$, $I = 3$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>measure</th>
<th>$\rho$ (density of views)</th>
<th>$10^{-5}$</th>
<th>$10^{-4}$</th>
<th>$10^{-3}$</th>
<th>$10^{-2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVLSA ($P=100$)</td>
<td>metric1</td>
<td>16.843</td>
<td>17.349</td>
<td>16.912</td>
<td>16.785</td>
<td>17.612</td>
</tr>
<tr>
<td>time (min)</td>
<td>0.003</td>
<td>0.009</td>
<td>0.000</td>
<td>0.009</td>
<td>0.009</td>
<td>0.009</td>
</tr>
<tr>
<td>Proposed ($\mu=0.05$)</td>
<td>metric1</td>
<td>0.478</td>
<td>0.610</td>
<td>0.565</td>
<td>0.775</td>
<td>0.803</td>
</tr>
<tr>
<td>time (min)</td>
<td>3.798</td>
<td>5.425</td>
<td>3.765</td>
<td>24.182</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed ($\mu=1$)</td>
<td>metric2</td>
<td>0.942</td>
<td>1.034</td>
<td>0.944</td>
<td>1.265</td>
<td>1.294</td>
</tr>
<tr>
<td>time (min)</td>
<td>2.182</td>
<td>3.714</td>
<td>4.510</td>
<td>16.378</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed ($\mu=5$)</td>
<td>metric2</td>
<td>0.003</td>
<td>0.021</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>time (min)</td>
<td>1.735</td>
<td>2.714</td>
<td>3.725</td>
<td>13.447</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To conclude, we have considered large-scale MAX-VAR GCCA with structure-promoting regularization and designed a memory-efficient and computationally lightweight algorithm that easily incorporates various common regularization penalties. Our analysis shows that the algorithm converges to a KKT point for a variety of regularizations at a sublinear rate. When the classic MAX-VAR GCCA is considered, the algorithm approaches a global optimal solution at a linear rate. Simulations demonstrated the good scalability and the effectiveness of the proposed algorithm.
5. REFERENCES


