ONLINE CANONICAL CORRELATION ANALYSIS VIA RAYLEIGH-RITZ PROJECTIONS

Vassilis Kalantzis[†]*, Panagiotis A. Traganitis*[*]*, and Charilaos I. Kanatsoulis*[**]

[†]IBM Research, MIT-IBM Watson AI Lab, Yorktown Heights, NY, USA
[*]Dept. of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA
[**]Dept. of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

2024 58th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-5405-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEEECONF60004.2024.10942937

ABSTRACT

Canonical correlation analysis (CCA) is a classical statistical tool that enables processing of multiview data in a plethora signal processing, machine learning and data mining applications, by identifying a common linear subspace from the available data views. Most algorithms tackling the CCA task require all the data per view to be available. Nevertheless, in many cases, data are not available in batch form and may arrive in streaming fashion. This work puts forth a novel, computationally efficient, projection based method for identifying and updating the common subspace on-the-fly, as new data arrive, while retaining its' fidelity. Preliminary numerical tests, on synthetic and real data benchmarks, showcase the potential of the proposed method.

Index Terms— Canonical Correlation Analysis, Online Analysis, Rayleigh-Ritz Projection

1. INTRODUCTION

Every single day, data are collected from multitudes of heterogeneous sensors, and generated by networked devices with different capabilities. While these sensors or devices may be observing the same physical phenomenon, the data they generate may not be immediately compatible, due to the differences of these devices. Processing, learning, and drawing inference, in such a scenario, calls for approaches that are able to capitalize on the multimodal nature of the data.

Canonical Correlation Analysis (CCA) is a powerful statistical tool that seeks a low-dimensional representation of two random vectors that maximizes their correlation coefficient [1, 2]. From an algebraic perspective, CCA extracts a common latent structure of a set of entities observed in two different feature domains, which are usually referred as the 'views' of the entities [3]. For example, the observed data may consist of video recordings of a phenomenon, in one view, and textual semantic descriptions of the same phenomenon, in another view. CCA finds applications in various fields, including but not limited to signal processing [4, 5, 6, 7, 8], machine learning [9, 10, 11, 12], natural language processing [13], data mining [14, 15], and bioinformatics [16].

CCA can be optimally solved via generalized eigenvalue decomposition [2]. However, this solution computes the square root decomposition of the matrix-views, which in general has cubic complexity and is computationally intensive for multi-dimensional data. Furthermore, CCA updates necessitate storing dense inverse-covariance matrices, leading to impractical memory requirements in various real-world scenarios. When dealing with data that arrive in a streaming fashion, the aforementioned complexity and memory issues are exacerbated, as the common latent representation has to be computed from scratch with every new datum.

Most approaches tackling the computational issues of CCA, both in batch and online scenaria, are gradient-based [17, 18, 19]. More recently, methods based on stochastic approximation and stochastic optimization methods were advocated in [20] and [21], respectively. In addition, an online and distributed version of CCA, with sparsity constraints was presented in [22].

In this paper, we capitalize on the algebraic perspective of CCA and propose a novel method to perform CCA in an online fashion. Leveraging the Rayleigh-Ritz approximation procedure, the proposed algorithm can dynamically update the canonical components and common subspace between two views of data. Besides data arriving online, the proposed method readily applies to scenaria where the data matrices do not fit in memory and have to be fetched in small batches. Compared to the prior art, the proposed algorithm deals with the eigendecomposition of the CCA objective directly, instead of relying on (inexact) gradient computations.

Notation: Lowercase bold letters, $\mathbf{x}$, denote vectors, uppercase bold letters, $\mathbf{X}$, represent matrices, and calligraphic uppercase letters, $\mathcal{X}$, stand for sets. The (i,j)-th entry of matrix $\mathbf{X}$ is denoted by $[\mathbf{X}]_{i,j}$. '$^\top$' denotes matrix or vector transpose, and $\mathrm{Ran}(\mathbf{X})$ and $\mathrm{Rank}(\mathbf{X})$ denote the rangespace and rank of $\mathbf{X}$, respectively, and $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. $\mathbf{I}$ and $\mathbf{0}$ the identity and all-zeroes matrices of appropriate dimension, respectively. Finally, the term "i-th leading eigenpair" of a matrix $\mathbf{X}$ will refer to its i-th algebraically largest eigenvalue and its corresponding (normalized) eigenvector.

2. PROBLEM STATEMENT AND PRELIMINARIES

Consider a dataset consisting of N data that is provided in two views, with $\mathbf{x}_i^{(1)}$ denoting the i-th $d_1 \times 1$ vector orresponding to the first view, and $\mathbf{x}_i^{(2)}$ denoting the i-th $d_2 \times 1$ vector corresponding to the second view. We collect all N vectors from the first and second view in the $N \times d_1$ matrix $\mathbf{X}^{(1)} = [\mathbf{x}_1^{(1)\top}, \ldots, \mathbf{x}_N^{(1)\top}]$ and $N \times d_2$ matrix $\mathbf{X}^{(2)} = [\mathbf{x}_1^{(2)\top}, \ldots, \mathbf{x}_N^{(2)\top}]$, respectively.

CCA seeks $d_1 \times k$ and $d_2 \times k$ projection matrices $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$, respectively, such that the resulting projected data $\mathbf{y}_n^{(1)} = \mathbf{P}^{(1)}\mathbf{x}_n^{(1)}$ and $\mathbf{y}_n^{(2)} = \mathbf{P}^{(2)}\mathbf{x}_n^{(2)}$ are maximally correlated, with respect to (w.r.t.) the Pearson correlation coefficient, for all $n = 1, \ldots, N$.

The classical CCA formulation [23, 24] can be cast as the fol-

Emails: vkal@ibm.com, traganit@msu.edu, kanac@seas.upenn.edu
The work of P. Traganitis is supported by NSF grant 2312546.

lowing optimization problem

$$\max_{\mathbf{P}^{(1)},\mathbf{P}^{(2)}} \operatorname{Tr}\left(\mathbf{P}^{(1)T}\mathbf{X}^{(1)T}\mathbf{X}^{(2)}\mathbf{P}^{(2)}\right) \quad \text{s.t. } \mathbf{P}^{(i)T}\mathbf{X}^{(i)T}\mathbf{X}^{(i)}\mathbf{P}^{(i)} = \mathbf{I}_k,\ i = 1, 2, \tag{1}$$

where $\mathbf{X}^{(n)} \in \mathbb{C}^{N\times d_i}$ is the i-th view containing N entities measured in a d_i-dimensional feature space, $\mathbf{P}^{(i)} \in \mathbb{C}^{d_i\times k}$ is a matrix that projects $\mathbf{X}^{(i)}$ into a subspace of dimension k and $\mathbf{I}_k$ represents the $k \times k$ identity matrix. Indeed one can observe that (1) maximizes the correlation between the two views of the data, under the constraint $\mathbf{P}^{(i)T}\mathbf{X}^{(i)T}\mathbf{X}^{(i)}\mathbf{P}^{(i)} = \mathbf{I}_k$, which ensures that no trivial solutions are obtained.

To solve the aforementioned optimization problem consider the covariance matrices $\mathbf{C}_{\mathbf{X_1}} = \mathbf{X}^{(1)T}\mathbf{X}^{(1)}$ and $\mathbf{C}_{\mathbf{X_2}} = \mathbf{X}^{(2)T}\mathbf{X}^{(2)}$ and assume that $\operatorname{Rank}(\mathbf{C}_{\mathbf{X_1}}) = \operatorname{Rank}(\mathbf{C}_{\mathbf{X_2}}) = N$. Letting $\mathbf{X}^{(i)} = \mathbf{U}^{(i)}\mathbf{\Sigma}^{(i)}\mathbf{V}^{(i)T}$ denote the Singular Value Decomposition (SVD) of $\mathbf{X}^{(i)}$, we define the matrix

$$\mathbf{Z}^{(i)} = \left(\mathbf{X}^{(i)T}\mathbf{X}^{(i)}\right)^{1/2}\mathbf{P}^{(i)} = \mathbf{V}^{(i)}\mathbf{\Sigma}^{(i)}\mathbf{V}^{(i)^T}\mathbf{P}^{(i)}. \tag{2}$$

Solving (2) with respect to $\mathbf{P}^{(i)}$, leads to

$$\mathbf{P}^{(i)} = \left(\mathbf{X}^{(i)T}\mathbf{X}^{(i)}\right)^{-1/2}\mathbf{Z}^{(i)} = \mathbf{V}^{(i)}\mathbf{\Sigma}^{(i)^{-1}}\mathbf{V}^{(i)T}\mathbf{Z}^{(i)}. \tag{3}$$

Using (3) the optimization problem in (1) can then be written as

$$\max_{\mathbf{Z}^{(1)},\mathbf{Z}^{(2)}} \operatorname{Tr}\left(\mathbf{Z}^{(1)\top}{\mathbf{C}_{\mathbf{X_1}}}^{-1/2}\mathbf{C}_{\mathbf{X_1X_2}}{\mathbf{C}_{\mathbf{X_2}}}^{-1/2}\mathbf{Z}^{(2)}\right) \quad \text{s.t. } \mathbf{Z}^{(i)\top}\mathbf{Z}^{(i)} = \mathbf{I}_k,\ i = 1, 2, \tag{4}$$

where $\mathbf{C}_{\mathbf{X_1X_2}} = \mathbf{X}^{(1)\top}\mathbf{X}^{(2)} = {\mathbf{C}_{\mathbf{X_2X_1}}}^{\top}$ is the cross-covariance matrix. The problem in (4) is equivalent to

$$\max_{\mathbf{Z}^{(1)},\mathbf{Z}^{(2)}} \operatorname{Tr}\left(\mathbf{Z}^{(1)T}\mathbf{V}^{(1)}\mathbf{U}^{(1)T}\mathbf{U}^{(2)}\mathbf{V}^{(2)T}\mathbf{Z}^{(2)}\right) \quad \text{s.t. } \mathbf{Z}^{(i)T}\mathbf{Z}^{(i)} = \mathbf{I}_k,\ i = 1, 2. \tag{5}$$

One way to solve this problem is to compute the top k singular vectors of the so-called normalized cross-covariance

$$\mathbf{T} = {\mathbf{C}_{\mathbf{X_1}}}^{-1/2}\mathbf{C}_{\mathbf{X_1X_2}}{\mathbf{C}_{\mathbf{X_2}}}^{-1/2} = \mathbf{V}^{(1)}\mathbf{U}^{(1)T}\mathbf{U}^{(2)}\mathbf{V}^{(2)T}.$$

Alternatively, invoking the Cauchy-Schwarz inequality, we observe that the problem in (4) is maximized when the columns of the matrices $\mathbf{Z}^{(1)}$ and $\mathbf{T}\mathbf{Z}^{(2)}$ are co-linear, i.e.,

$$\begin{aligned}\mathbf{Z}^{(1)} &= {\mathbf{C}_{\mathbf{X_1}}}^{-1/2}\mathbf{C}_{\mathbf{X_1X_2}}{\mathbf{C}_{\mathbf{X_2}}}^{-1/2}\mathbf{Z}^{(2)}\\ &= \mathbf{V}^{(1)}\mathbf{U}^{(1)T}\mathbf{U}^{(2)}\mathbf{V}^{(2)T}\mathbf{Z}^{(2)}.\end{aligned}$$

Using this fact, the canonical components can be recovered via

$$\max_{\mathbf{Z}^{(2)}} \operatorname{Tr}\left(\mathbf{Z}^{(2)T}\mathbf{V}^{(2)}\mathbf{U}^{(2)T}\mathbf{U}^{(1)}\mathbf{U}^{(1)T}\mathbf{U}^{(2)}\mathbf{V}^{(2)T}\mathbf{Z}^{(2)}\right) \quad \text{s.t. } \mathbf{Z}^{(2)T}\mathbf{Z}^{(2)} = \mathbf{I}_k, \tag{6}$$

which can be solved by setting $\mathbf{Z}^{(2)}$ equal to the matrix formed by the k principal eigenvectors of the matrix

$$\begin{aligned}\mathbf{M} &= {\mathbf{C}_{\mathbf{X_2}}}^{-1/2}\mathbf{C}_{\mathbf{X_2X_1}}{\mathbf{C}_{\mathbf{X_1}}}^{-1}\mathbf{C}_{\mathbf{X_1X_2}}{\mathbf{C}_{\mathbf{X_2}}}^{-1/2}\\ &= \mathbf{V}^{(2)}\mathbf{U}^{(2)T}\mathbf{U}^{(1)}\mathbf{U}^{(1)T}\mathbf{U}^{(2)}\mathbf{V}^{(2)T}.\end{aligned} \tag{7}$$

These k eigenvectors can be computed via a matrix-free, symmetric eigenvalue solver such as the Lanczos method [25]. Let now $\mathrm{MV}(\mathbf{X})$ denote the computational cost to multiply a matrix $\mathbf{X}$ with a vector of conforming size. Then, applying the Lanczos method to the matrix M yields an asymptotic computational cost $\mathcal{O}\left(\left[\mathrm{MV}(\mathbf{U}^{(1)}) + \mathrm{MV}(\mathbf{U}^{(2)}) + \mathrm{MV}(\mathbf{V}^{(2)})\right]k + d_2k^2\right)$ [26].

Finally, the $N \times k$ common subspace $\mathbf{G}$ can then be recovered using $\mathbf{Z}^{(2)}$ as

$$\begin{aligned}\mathbf{G} &= \mathbf{X}^{(2)}\mathbf{P}^{(2)} = \mathbf{X}^{(2)}\left(\mathbf{X}^{(2)T}\mathbf{X}^{(2)}\right)^{-1/2}\mathbf{Z}^{(2)}\\ &= \mathbf{U}^{(2)}{\mathbf{V}^{(2)}}^{T}\mathbf{Z}^{(2)}.\end{aligned} \tag{8}$$

3. ONLINE CCA

Consider now a setting where the data from the two views are dynamically updated with the addition of new data samples or features. In this case, as the data matrices change, so does the respective matrix $\mathbf{M}$ in (7). A naive approach to update the canonical components is to recompute the k dominant eigenvectors of $\mathbf{M}$ from scratch, e.g., by applying the Lanczos scheme. This approach, however, does not take advantage of previous computational efforts. In this section we consider a method to update the canonical components subject to periodic augmentation of an initial set of data while re-using the previously computed canonical components. The proposed scheme is based on the Rayleigh-Ritz (RR) procedure, discussed next.

3.1. The Rayleigh-Ritz approximation procedure

The RR procedure aims to approximate a portion of the eigenvalues and eigenvectors of a matrix $\mathbf{M}$ via projection onto a carefully chosen ansatz subspace $\mathcal{Q}$, which, ideally, contains the invariant subspace associated with the eigenvalues of interest [27]. The RR procedure is effectively a dimensionality reduction technique which in lieu of computing eigenpairs of $\mathbf{M}$ instead computes eigenpairs of the matrix $\mathbf{Q}^T\mathbf{M}\mathbf{Q}$. Here the matrix $\mathbf{Q}$ represents a basis of the low-dimensional subspace $\mathcal{Q}$. The eigenpairs of $\mathbf{M}$ can then be approximated by linear combinations of the columns of $\mathbf{Q}$.

Specifically, let $\{(\tau_i, \mathbf{h}_i)\}_{i=1}^k$, denote the k leading eigenpairs of the matrix $\mathbf{Q}^T\mathbf{M}\mathbf{Q}$, where τ_i are the eigenvalues and $\mathbf{h}_i$ the corresponding eigenvectors. Then, the the i-th leading eigenpair of the matrix $\mathbf{M}$ is approximated by the so-called i-th leading Ritz pair $(\tau_i, \mathbf{Q}\mathbf{h}_i)$. In fact, when the subspace $\mathcal{Q}$ contains an invariant subspace associated with the k dominant eigenvalues of $\mathbf{M}$, the Ritz pairs are equal, up to roundoff error, to the true k dominant eigenpairs of $\mathbf{M}$ [28, Section 11].

3.2. A scheme for updating the canonical components

Suppose that at time-step 't' the data $\mathbf{X}_t^{(1)}, \mathbf{X}_t^{(2)}$ are available and consequently the k dominant eigenvectors of the matrix $\mathbf{M}_t$ form the canonical components of the two views [see Sec. 2]. Then at time-step '$t+1$', n_{t+1} new data arrive. The new data are similarly split into two views $\tilde{\mathbf{X}}_{t+1}^{(1)}$ and $\tilde{\mathbf{X}}_{t+1}^{(2)}$ respectively. Then in each view the matrices $\mathbf{X}_t^{(1)}$ and $\mathbf{X}_t^{(2)}$ become submatrices of the matrices $\mathbf{X}_{t+1}^{(1)}$ and $\mathbf{X}_{t+1}^{(2)}$, i.e.

$$\mathbf{X}_{t+1}^{(1)} = \begin{bmatrix}\mathbf{X}_t^{(1)}\\ \tilde{\mathbf{X}}_{t+1}^{(1)}\end{bmatrix}, \quad \mathbf{X}_{t+1}^{(2)} = \begin{bmatrix}\mathbf{X}_t^{(2)}\\ \tilde{\mathbf{X}}_{t+1}^{(2)}\end{bmatrix}.$$

Following the computations of Section 2 yields a new matrix $\mathbf{M}_{t+1}$ whose k dominant eigenvectors now form the canonical components of the augmented views.

Algorithm 1 Online CCA (at least one update occurs).

Input: $\mathbf{X}_1^{(1)}, \mathbf{X}_1^{(2)}$
Output: k leading eigenpairs of the matrix $\mathbf{M}_{t+1}$
▷ Set $t \leftarrow 1$ and construct the projection matrix $\mathbf{Q}_t$
do
  ▷ Update $\mathbf{X}_t^{(1)}, \mathbf{X}_t^{(2)}$ to $\mathbf{X}_{t+1}^{(1)}, \mathbf{X}_{t+1}^{(2)}$
  ▷ Build the projection matrix $\mathbf{Q}_{t+1}$ [See Sec.3.3]
  ▷ Compute the eigenpairs $\{(\tau_{t+1,i}, \mathbf{h}_{t+1,i})\}_{i=1}^{k}$ of $\mathbf{Q}_{t+1}^T \mathbf{M}_{t+1} \mathbf{Q}_{t+1}$
  ▷ Set $\mathbf{Z}_{t+1}^{(2)} = [\mathbf{Q}_{t+1}\mathbf{h}_{t+1,1}, \ldots, \mathbf{Q}_{t+1}\mathbf{h}_{t+1,k}]$
  ▷ Update $t \leftarrow t+1$
while there exist updates

Algorithm 1 summarizes the proposed framework. Upon arrival of the new data and formation of $\mathbf{X}_{t+1}^{(1)}$ and $\mathbf{X}_{t+1}^{(2)}$, the thin SVD $\mathbf{X}_{t+1}^{(i)} = \mathbf{U}_{t+1}^{(i)} \mathbf{\Sigma}_{t+1}^{(i)} \mathbf{V}_{t+1}^{(i)\top}$, $i = 1, 2$ is computed. The thin SVD of $\mathbf{X}_{t+1}^{(i)}$ does not have to be computed from scratch; instead, the SVD of the matrix $\mathbf{X}_t^{(i)}$ can be used as a warm-start mechanism [29, 30, 31].

With the SVD of $\mathbf{X}_{t+1}^{(1)}$ and $\mathbf{X}_{t+1}^{(2)}$ at hand, the scheme proceeds with the approximation of the k dominant eigenvectors of the matrix

$$\mathbf{M}_{t+1} \;=\; \mathbf{V}_{t+1}^{(2)} \mathbf{U}_{t+1}^{(2)T} \mathbf{U}_{t+1}^{(1)} \mathbf{U}_{t+1}^{(1)T} \mathbf{U}_{t+1}^{(2)} \mathbf{V}_{t+1}^{(2)T}.$$

Rather than naively applying the Lanczos method to $\mathbf{M}_{t+1}$, the novelty of Algorithm 1 lies on applying a Rayleigh-Ritz approximation mechanism which exploits the previously computed eigenvectors of the matrix $\mathbf{M}_t$. The next section derives the projection subspace when the number of samples increases dynamically.

3.3. Setting up the projection subspace

A starting point to set the projection matrix $\mathbf{Q}_{t+1}$ is to consider the k dominant Ritz vectors of the matrix $\mathbf{M}_t$, i.e., $\mathbf{Q}_{t+1} = [\mathbf{Q}_t\mathbf{h}_{t,1}, \ldots, \mathbf{Q}_t\mathbf{h}_{t,k}]$. This choice can be efficient when $\mathbf{M}_t \approx \mathbf{M}_{t+1}$, however it ignores any information introduced during time-step '$t+1$'. To overcome this limitation, one can consider enriching the canonical components obtained at time-step 't' with information obtained from the newly added samples in matrices $\mathbf{X}_{t+1}^{(1)}$ and $\mathbf{X}_{t+1}^{(2)}$. The following proposition outlines the conditions for exactly recovering the eigenvectors of $\mathbf{M}_{t+1}$ via the RR procedure.

Proposition 3.1. *Let* $k \leq \mathrm{Rank}(\mathbf{M}_{t+1})$ *and*

$$\mathrm{Ran}(\mathbf{Q}_{t+1}) = \mathrm{Ran}\left(\mathbf{V}_{t+1}^{(2)} \mathbf{U}_{t+1}^{(2)T} \mathbf{U}_{t+1}^{(1)}\right).$$

Then, the $\min(k, \mathrm{Rank}(\mathbf{M}_{t+1}))$ *dominant Ritz pairs* $(\tau_{t,i}, \mathbf{Q}_t\mathbf{h}_{t,i})$ *are equal to the* $\min(k, \mathrm{Rank}(\mathbf{M}_{t+1}))$ *dominant eigenpairs of the matrix* $\mathbf{M}_{t+1}$.

Proof. The proof follows directly by noticing that

$$\mathrm{Ran}\left(\mathbf{V}_{t+1}^{(2)} \mathbf{U}_{t+1}^{(2)T} \mathbf{U}_{t+1}^{(1)}\right) \equiv \mathrm{Ran}(\mathbf{M}_{t+1}),$$

and recalling that any eigenvector associated with a non-zero eigenvalue must lie in the matrix column space. □

Invoking Prop. 3.1 and noting that $\mathrm{Ran}\left(\mathbf{V}_{t+1}^{(2)} \mathbf{U}_{t+1}^{(2)T} \mathbf{U}_{t+1}^{(1)}\right) \subseteq \mathrm{Ran}(\mathbf{V}_{t+1}^{(2)})$, yields the following corollary.

Corollary 1. *Let* $k \leq d_2$. *Then, if* $\mathbf{Q}_{t+1} = \mathbf{V}_{t+1}^{(2)}$, *the* k *dominant Ritz pairs* $(\tau_{t+1,i}, \mathbf{Q}_{t+1}\mathbf{h}_{t+1,i})$ *are equal to the* k *dominant eigenpairs of the matrix* $\mathbf{M}_{t+1}$.

Based on Corollary 1 the projection matrix $\mathbf{Q}_{t+1}$ can be chosen as the $d_2 \times \mathrm{Rank}(\mathbf{M}_{t+1})$ matrix $\mathbf{V}_{t+1}^{(2)}$ that contains the right singular vectors of $\mathbf{X}_{t+1}^{(2)}$. Nevertheless, using $\mathbf{V}_{t+1}^{(2)}$ directly does not reduce the dimension of the projection subspace as it is equivalent to a similarity transformation. Instead, the idea considered in this work is to extend the projection subspace from the previous time-step by adding k dominant right singular vectors of the matrix $\mathbf{V}_{t+1}^{(2)}$, i.e.,

$$\mathbf{Q}_{t+1} = \mathrm{Orth}\left(\left[\mathbf{Q}_t\mathbf{h}_{t,1}, \ldots, \mathbf{Q}_t\mathbf{h}_{t,k}, \mathbf{V}_{t+1,1:k}^{(2)}\right]\right),$$

where $\mathbf{V}_{t+1,1:k}^{(2)}$ is the $d_2 \times k$ matrix containing the right singular vectors associated with the k dominant singular values of $\mathbf{X}_{t+1}^{(2)}$, and $\mathrm{Orth}(\cdot)$ is a matrix operator that orthogonalizes its argument, via e.g. the Gram-Schmidt procedure. Using $\mathbf{Q}_{t+1}$, one can obtain the k dominant eigenpairs $\{(\tau_{t+1,i}, \mathbf{h}_{t+1,i})\}_{i=1}^{k}$ of $\mathbf{Q}_{t+1}^\top \mathbf{M}_{t+1} \mathbf{Q}_{t+1}$, and the matrix of interest $\mathbf{Z}_{t+1}^{(2)} = [\mathbf{Q}_{t+1}\mathbf{h}_{t+1,1}, \ldots, \mathbf{Q}_{t+1}\mathbf{h}_{t+1,k}]$. Upon computing $\mathbf{Z}_{t+1}^{(2)}$, $\mathbf{Z}_{t+1}^{(1)}$ can be recovered via (6). The common subspace between the two views at time $'t+1'$ is given by (8). This procedure is repeated as more new data arrive. The performance of Alg. 1 will be evaluated in the next section.

4. NUMERICAL TESTS

The proposed approach is benchmarked using synthetically generated and real data, all tests were conducted using MATLAB, and results represent averages over 10 independent Monte Carlo runs. The synthetic data were created using the generative model in [3], where each noiseless view is given by $\check{\mathbf{X}}^{(i)} = [\mathbf{A}\mathbf{B}^{(i)}]\mathbf{C}^{(i)\top}$, with $\mathbf{A}$ being the $N \times k$ common subspace between the two views, and $\mathbf{B}^{(i)} \in \mathbb{C}^{N \times \ell_n}$ and $\mathbf{C}^{(i)} \in \mathbb{C}^{d_2 \times k + \ell_n}$ matrices, for some $\ell_n \in \mathbb{N}$ pertaining to each view. Here, $N = 2,000$, $k = 15$, $d_1 = d_2 = 1015$, and the entries of $\mathbf{A}, \mathbf{B}^{(i)}, \mathbf{C}^{(i)}$ are drawn from the standard normal distribution. After generating the noiseless matrices $\check{\mathbf{X}}^{(i)}$ for each view, additive white Gaussian noise is added, i.e. $\mathbf{X}^{(i)} = \check{\mathbf{X}}^{(i)} + \mathbf{N}$, where $[\mathbf{N}]_{i,j} \sim \mathcal{N}(0, \sigma^2)$ for all i, j. The variance σ^2 is chosen so that the signal-to-noise ratio is approximately 15dB.

The real dataset considered is the Mediamill dataset [32], which consists of $N = 10,000$ observations of videos with paired commentary. One view of the data consists of $d_1 = 120$ features extracted from the videos, while the second view consists of $d_2 = 100$ textual features representing semantics of what is portrayed in the video.

Figures 1-2 plot the angle formed per dominant eigenvector of $\mathbf{M}$, i.e. columns of $\mathbf{Z}^{(2)}$, computed with the entire dataset, and the output $\hat{\mathbf{Z}}^{(2)}$ of Alg. 1 at the end of the final time-step as k varies, i.e.

$$\mathrm{angle}(\mathbf{Z}^{(2)}, \hat{\mathbf{Z}}^{(2)}) = \|\mathbf{W} - \hat{\mathbf{W}}\|_2$$

where $\mathbf{W}$ and $\hat{\mathbf{W}}$ are the projections onto $\mathbf{Z}^{(2)}$ and $\hat{\mathbf{Z}}^{(2)}$, respectively. In Fig. 1 we consider only one time-step, where for each dataset we hide a percentage of $\tau \in \mathbb{R}$ bottom rows of the matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. These rows are then introduced as new data. Naturally, introducing fewer rows leads to a better accuracy of our

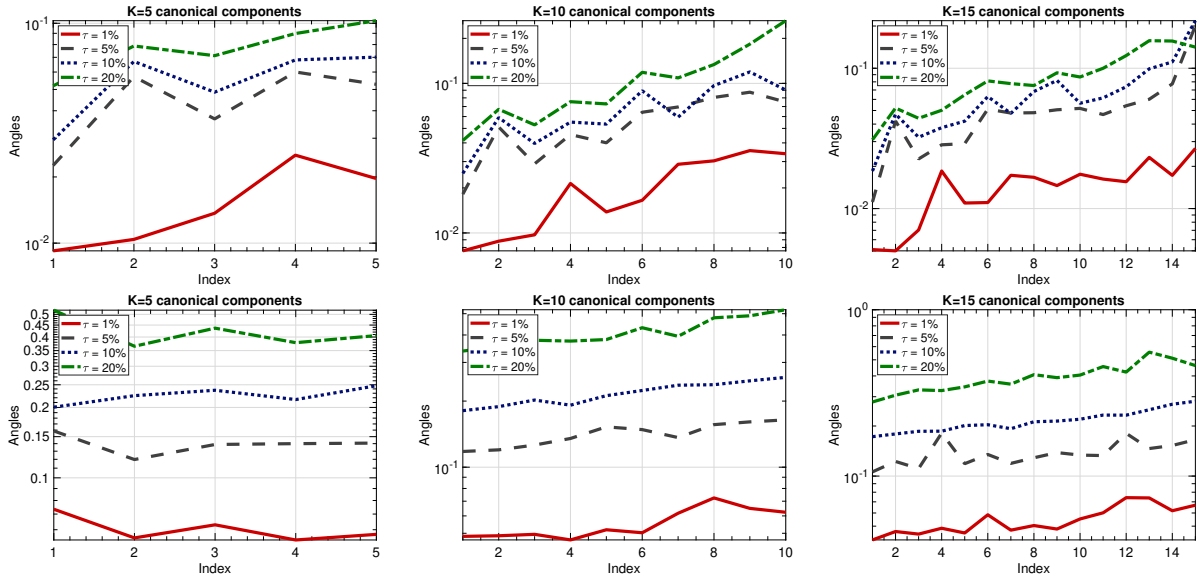

Fig. 1. Angle between the i-th dominant eigenvector and the invariant subspace returned by Algorithm 1 (single time-step case). Top row: Mediamill dataset [32]. Bottom row: synthetic dataset.

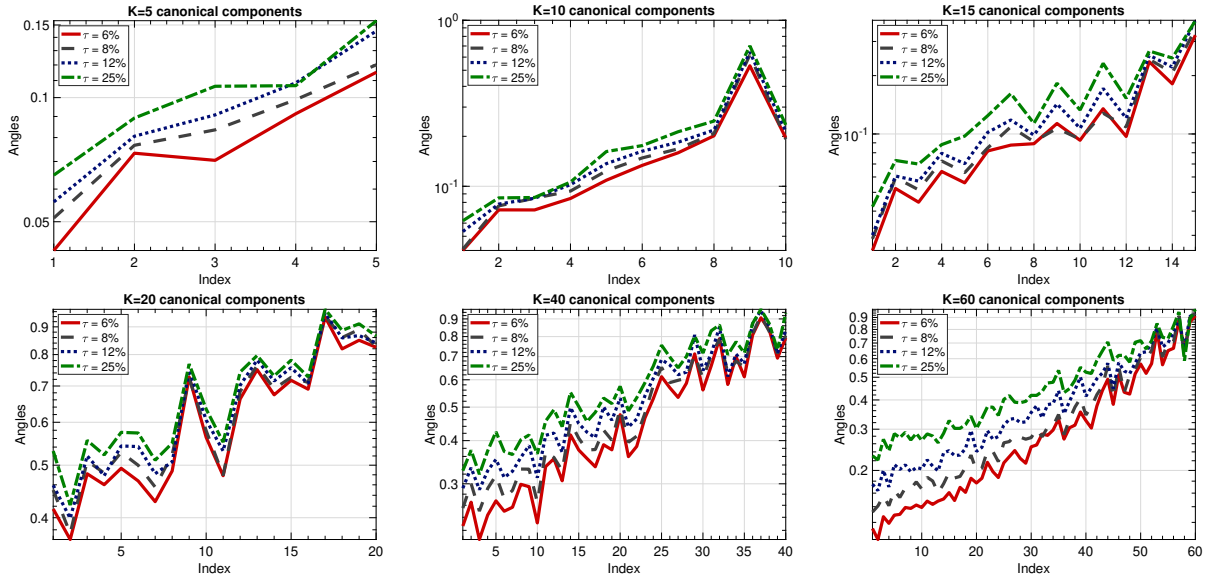

Fig. 2. Angle between the i-th dominant eigenvector and the invariant subspace returned by Algorithm 1, for the case where $2, 4, 6$ and 8 time-steps are considered. Top row: Mediamill dataset [32]. Bottom row: synthetic dataset.

scheme since the invariant subspace computed at the initial stage is more similar to that obtained after a single update. This is true both for the Mediamill and synthetic datasets. Fig. 2 extends the previous experiment by introducing multiple time-steps. This time we hide exactly half of the rows of the matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, and the hidden rows are introduced in $2, 4, 6$ and 8 time-steps, with each time-step introducing approximately 24%, 12%, 8%, and 6%, of the total dataset rows. As in the previous experiment, updates with fewer data lead to higher overall accuracy. On the other hand, more updates indicate that more SVD updates are required, and thus there is an accuracy-speed trade-off.

5. CONCLUSIONS

This work introduced a novel method for performing canonical correlation analysis when data arrive in a streaming fashion. The proposed algorithm capitalizes on the Rayleigh-Ritz approximation method to efficiently and accurately update the canonical components. Future work will include extending the proposed algorithm to more than two views, extensive tests on real data and comparisons with competing alternatives, as well as a rigorous performance analysis.

6. REFERENCES

[1] H. Hotelling, "Relations between two sets of variants," *Biometrika*, vol. 28, no. 3–4, pp. 321–377, 1936.

[2] G. H. Golub and H. Zha, "The canonical correlations of matrix pairs and their numerical computation," *IMA Vol. Math. Appl,*, vol. 69, pp. 27–49, 1995.

[3] M. Sørensen, C. I. Kanatsoulis, and N. D. Sidiropoulos, "Generalized canonical correlation analysis: A subspace intersection approach," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2452–2467, 2021.

[4] J. Vía, I. Santamaría, and J. Pérez, "Deterministic CCA-based algorithms for blind equalization of FIR-MIMO channels," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, July 2007.

[5] S. Van Vaerenbergh, J. Vía, and I. Santamaría, "Blind identification of SIMO Wiener systems based on kernel canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 61, pp. 2219–2230, May 2013.

[6] R. Arora and K. Livescu, "Multi-view learning with supervision for transformed bottleneck features," in *Proc. ICASSP 2014*, 2014, pp. 2499–2503.

[7] J. Manco-Vásquez, S. Van Vaerenbergh, J. Vía, and I. Santamaría, "Kernel canonical correlation analysis for robust cooperative spectrum sensing in cognitive radio networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, 2014.

[8] M. Ibrahim and N. D. Sidiropoulos, "Reliable detection of unknown cell-edge users via canonical correlation analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4170–4182, Jun 2020.

[9] A. Vinokourov, J. J. Shawe-Taylor, and N. Cristianini, "Inferring a semantic representation of text via cross-language correlation analysis," in *Proc. NIPS 2003, December 11-13, 2003, Whistler, British Columbia, Canada*.

[10] 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, Dec. 2004.

[11] P. Dhillon, D. Foster, and L. Ungar, "Multi-view learning of word embeddings via CCA," in *Proc. NIPS 2011, December 12-17, 2011, Granada, Spain*.

[12] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 17–19 Jun 2013, pp. 1247–1255.

[13] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Hong, "Structured sumcor multiview canonical correlation analysis for large-scale data," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 306–319, Jan 2019.

[14] S. Bickel and T. Scheffer, "Multi-view clustering.," in *ICDM*, 2004, vol. 4, pp. 19–26.

[15] P. Rastogi, B. Van Durme, and R. Arora, "Multiview lsa: Representation learning via generalized cca.," in *HLT-NAACL*, 2015, pp. 556–566.

[16] J. C. Vasquez-Correa, J. R. Orozco-Arroyave, R. Arora, E. Nöth, N. Dehak, H. Christensen, F. Rudzicz, T. Bocklet, M. Cernak, H. Chinaei, et al., "Multi-view representation learning via gcca for multimodal analysis of parkinson" s disease," in *Proceedings of 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)*, 2017.

[17] 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.

[18] R. Ge, C. Jin, P. Netrapalli, A. Sidford, et al., "Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis," in *International Conference on Machine Learning*, 2016, pp. 2741–2750.

[19] Z. Ma, Y. Lu, and D. Foster, "Finding linear structure in large datasets with scalable canonical correlation analysis," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 07–09 Jul 2015, pp. 169–178.

[20] R. Arora, T. V. Marinov, P. Mianjy, and N. Srebro, "Stochastic approximation for canonical correlation analysis," in *Advances in Neural Information Processing Systems*, 2017.

[21] C. Gao, D. Garber, N. Srebro, J. Wang, and W. Wang, "Stochastic canonical correlation analysis," *Journal of Machine Learning Research*, vol. 20, no. 167, pp. 1–46, 2019.

[22] J. Chen and I. D. Schizas, "Online distributed sparsity-aware canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 3, pp. 688–703, 2016.

[23] G. H. Golub and H. Zha, *The canonical correlations of matrix pairs and their numerical computation*, Springer, 1995.

[24] 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.

[25] Y. Saad, *Numerical methods for large eigenvalue problems: revised edition*, SIAM, 2011.

[26] V. Kalantzis, "A domain decomposition rayleigh–ritz algorithm for symmetric generalized eigenvalue problems," *SIAM Journal on Scientific Computing*, vol. 42, no. 6, pp. C410–C435, 2020.

[27] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 2013.

[28] B. N. Parlett, *The symmetric eigenvalue problem*, SIAM, 1998.

[29] V. Kalantzis, G. Kollias, S. Ubaru, A. N. Nikolakopoulos, L. Horesh, and K. Clarkson, "Projection techniques to update the truncated SVD of evolving matrices with applications," in *International Conference on Machine Learning*, 2021, pp. 5236–5246.

[30] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear algebra and its applications*, vol. 415, no. 1, pp. 20–30, 2006.

[31] M. Moonen, P. Van Dooren, and J. Vandewalle, "A singular value decomposition updating algorithm for subspace tracking," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1015–1038, 1992.

[32] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings of the 14th ACM International Conference on Multimedia*, New York, NY, USA, 2006, p. 421–430.