Tensor Canonical Correlation Analysis

You-Lin Chen, Mladen Kolar, Ruey S. Tsay

Abstract

In this paper, we formulate the Canonical Correlation Analysis on tensor-valued data. We consider two efficient algorithms: Higher-order Power method and Alternating Least Squares. Their relation are carefully analyzed and local convergence properties are established via the theory of Lojasiewicz inequalities. Further, we present the inexact updating scheme and its error analysis, which allows us to use state-of-the-arts stochastic gradient descent algorithms for large-scale data. Experiments on several real data set show effectiveness and efficiency and demonstrate their superior potential.

1 Introduction

Canonical Correlation Analysis (CCA) [25], which is a multivariate analysis method, addresses the problem of correlating linear relationships between two multidimensional variables. The object of CCA, we also call 1DCCA in this paper, is projecting those two variables sets linearly onto a low-dimensional space where they are maximally correlated. CCA can be applied to many successful applications. For example, CCA is used in text and image retrieval [38], image clustering [26] and various fields [24]. In contrast to unsupervised learning such as Principal component analysis (PCA), CCA gets benefits from the label information [45] as supervised learning or other views of data (noisy, rotated, shifted, etc) and correlated side information as multi-view Learning [55].

Many extensions of CCA exist, and various data structures are considered. [46] discusses sparse CCA and provides several algorithms such as linearized alternating direction minimization method, TFOCS and semidefinite programming approach, and [47] uses Rayleigh Flow to formulate generalized eigenvalue problem and use gradient descent with truncation to get sparsity solution. For nonlinear extensions, kernel method aiming to extract the nonlinear relation via implicitly mapping data to a high-dimensional feature space can be easily applied to CCA by replacing usual inner product to kernel and covariance matrix to kernel matrix [24, 23, 18]. Moreover, other nonlinear transformations are considered to get more sophisticated relation of data. [3] utilizes neural networks to learn such nonlinear transformations, and [37] uses Lancasters theory for extending CCA to a nonparametric setting. [34] proposes randomized nonlinear CCA which uses random feature to construct a kernel matrix and can be view a low-rank approximation. [5, 41] interpret CCA as a latent variable model and applies EM algorithms to this problem. Following the idea of probabilistic CCA model, the variational inference is used in [42, 52]. However, involved nonlinear extensions of CCA usually have more complicated optimization problem and lack of rigorous convergence analysis.

On the other hand, although CCA can be solved by the generalized eigenproblem (See section 3 for more details), computing SVD or EVD for large covariance matrix and the square root of the inverse of the large covariance matrix are computation expensive. Finding an efficient optimization method for CCA is still an important problem, especially for large-scale data. Several algorithms for

large-scale data have been proposed recently. Most of them are based on the least square formula of CCA and stochastic optimization [20, 2, 14, 50, 31, 17]. For example, [51] use alternating least squares formulation of CCA and use stochastic gradient descent to solve this regularized least squares. They also propose the shift-and-invert procedure in the case of smell eigenvalue gap. [19] gives a statistical consistent analysis. [4] proposes an online solution for CCA using inexact matrix stochastic gradient and inexact matrix exponential gradient. [60] formulate the CCA on matrix manifolds and use the view of optimization on the Riemannian manifold to provide an adaptive algorithm. Notably, [36] provides Augmented Approximate Gradient (AppGrad) scheme which avoids computing the inverse of large matrices.

In this paper, a tensor extension of CCA for matrix-valued or tensor-valued data is presented. The native way to deal with tensor data is transforming the data into vector form and applying 1DCCA. The problem is that this result in increasing the dimension of data exponentially. To circumvent this difficulty and reduce the computation cost, we maintain the structure of tensor and use a multilinear map to formulate CCA problem. Solving the Tensor version of CCA is equivalent to tensor decomposition with more complicated data-dependent constraint. The high-order power method (HOPM) is a simple, effective and widely used optimization algorithm in tensor decomposition [29, 12, 13]. However, tensor decomposition is a non-convex optimization, so it suffers from the problem of a local optimum. Also, it is much more difficult to analyze the convergence property that has been intensely studied recently [48, 53, 56, 33, 21]. Since CCA have the more complicated constraint, we can not directly apply those result. In this paper, we propose HOPM with a simpler constraint which we call alternative least square (ALS) algorithm and verify their equivalence carefully. An easy convergence proof is presented without any explicit model assumption, which based on the elegant theory of analytical gradient flows and Lojasiewicz gradient inequality. The benefit of using this methodology is not only that we only require simple model-free assumptions but also that this convergence analysis can be applied to any stationary point. Moreover, we propose effective initialization which reduces the probability of trapping in local maximum and an inexact updating rule which allow us to use gradient descent method suitable for large-scale data as well as the error analysis.

1.1 Main Contributions

Our main contribution is the following

- A clear and formal definition of tensor Canonical Correlation Analysis is given, which related to low rank matrix factorization and tensor decomposition.
- We consider the power method and its variants and study their relation and convergence properties.
- The power method is equivalent to solving least square problem, which leads a efficient gradient based optimization algorithms. We give a rigorous error bound for this inexact updating.
- A effective initialization scheme that can reduce the probability of obtaining local optimums is developed.
- We apply our method to real data including air pollution data, electricity demand data and gene expression data, showing our method not only is more efficient and low computational cost but also can finding useful facts in practice.

2 Related works

There are several works closed relating to our work. Two-dimensional CCA (2DCCA) which works directly on matrix data is first proposed by [30]. Our work can be viewed as a natural extension of 2DCCA to tensor data. We propose a more efficient algorithm with rigorous theoretical verification. Many 2D version of dimension reduction method like PCA, PLS, LDA, are proposed [62, 57, 58, 59]. All of them can be treated as a tensor extension, and alternative minimization is the most frequently used optimization method. Both [28] and [35] discuss tensor setting of CCA. [28] consider a more complicated way to correlate different mode in tensor data, and [35] consider more than two views in CCA and have covariance tensor to optimize. None of them provide convergence analysis, and they have the different setting than us.

3 Preliminaries

3.1 Multilinear Algebra and Notation

In this section, we will briefly introduce useful notation and concepts of multilinear algebra. For more details, see the review paper [29]. A tensor is a multi-dimensional array and the order of a tensor is the number of dimensions, also called way or mode. For example, a vector is a first-order tensor, and a matrix is a second-order tensor. To distingish between scalars, vectors, matrices, and higher-order tensors, we use following convention: scalars are denoted by lower-case letter $(a, b, c, \dots; \alpha, \beta, \gamma, \dots)$, vectors are written as capitals $(A, B, \dots, X, Y)$, matrices correspond to bold-face capitals $(\mathbf{A}, \mathbf{B}, \dots, \mathbf{X}, \mathbf{Y})$, and tensors are written as calligraphic letters $(\mathcal{A}, \mathcal{B}, \dots, \mathcal{X}, \mathcal{Y})$. For an m-mode tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$, we let its $(i_1, \dots, i_m)$-th element be denoted by $x_{i_1 i_2 \dots i_m}$ or $(\mathcal{X})_{i_1 i_2 \dots i_m}$.

Now we define some useful operators in multilinear algebra. Matricization, also known as unfolding, is a process of transforming a tensor into a matrix. The mode-j matricization of a tensor $\mathcal{X}$ is denoted by $\mathcal{X}_{(a)}$ and arranges the mode-a fibers to be the columns of the resulting matrix. More specifically, tensor element $(i_1, \dots, i_m)$ maps to matrix element (i_a, j), where

$$j = 1 + \sum_{k=1, k \neq a}^{m} (i_k - 1) j_k \quad \text{with} \quad j_k = \prod_{q=1, q \neq a}^{k-1} i_q$$

We also denote vectorizating $\mathcal{X}$ by $\text{vec}(\mathcal{X})$. The Frobenius norm of the tensor $\mathcal{X}$ is defined by

$$\|\mathcal{X}\|_F^2 = \langle \mathcal{X}, \mathcal{X} \rangle$$

where $\langle \cdot, \cdot \rangle$ is the inner product defined on two tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$, $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ given by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{d_1} \cdots \sum_{i_m=1}^{d_m} x_{i_1 i_2 \dots i_m} y_{i_1 i_2 \dots i_m}$$

The mode-k product of a tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ with a matrix $\mathbf{A} \in \mathbb{R}^{a \times d_k}$ is a tensor of size $d_1 \times \cdots \times d_{k-1} \times a \times d_{k+1} \cdots \times d_m$ defined as

$$(\mathcal{X} \times_k \mathbf{A})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_m} = \sum_{i_k=1}^{d_k} x_{i_1 i_2 \dots i_m} a_{j i_k}$$

The outer product of vectors $U_1 \in \mathbb{R}^{d_1}, \ldots, U_m \in \mathbb{R}^{d_m}$ is a m-order tensor definde by

$$(U_1 \circ \cdots \circ U_m)_{i_1 i_2 \ldots i_m} = (U_1)_{i_1} \ldots (U_m)_{i_m}$$

and Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ is an $mp \times nq$ matrix defined by

$$\mathbf{A} \otimes \mathbf{B} = (a_{ij}\mathbf{B})_{mp \times nq}$$

We call $\mathcal{X}$ is a rank-one tensors if there exist vectors $X_1, \ldots, X_d$ such that

$$\mathcal{X} = X_1 \circ \cdots \circ X_d$$

3.2 1D CCA

Consider two multivariate random vectors $X \in \mathbb{R}^{d_x}, Y \in \mathbb{R}^{d_y}$. The CCA can be formed as following:

$$\max_{U \in \mathbb{R}^{d_x}, V \in \mathbb{R}^{d_y}} \operatorname{corr}(U^\top X, V^\top Y) = \max_{U \in \mathbb{R}^{d_x}, V \in \mathbb{R}^{d_y}} \frac{\operatorname{cov}(U^\top X, V^\top Y)}{\sqrt{\operatorname{var}(U^\top X)\operatorname{var}(V^\top Y)}}$$

Provided that two random vector are of zero mean, CCA also can be posted by following the optimization problem:

$$\max_{U,V} \frac{\mathbf{E}[U^\top X Y^\top V]}{\sqrt{\mathbf{E}[U^\top X]^2 \, \mathbf{E}[V^\top Y]^2)}}$$

which is equivalent to constrained form:

$$\max_{U,V} U^\top \Sigma_{XY} V \text{ s.t. } U^\top \Sigma_{XX} U = 1 = V^\top \Sigma_{YY} V$$

where $\Sigma_{XY} = \mathbf{E}[XY^\top]$, $\Sigma_{XX} = \mathbf{E}[XX^\top]$ and $\Sigma_{YY} = \mathbf{E}[YY^\top]$. This can be solved by different fomula which relate to eigenproblem. By standard technique of Lagrangian multiplier, we can get the U, V by the following generalized eigenvalue problem

$$\begin{bmatrix} 0 & \Sigma_{XY} \\ \Sigma_{YX} & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \lambda \begin{bmatrix} \Sigma_{XX} & 0 \\ 0 & \Sigma_{YY} \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} \tag{1}$$

where $\Sigma_{YX} = \Sigma_{XY}^\top$. Provided Σ_{YY} is invertible, we have

$$V = \Sigma_{YY}^{-1} \Sigma_{YX} U / \lambda \tag{2}$$

Substituting to (1) implies

$$\Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} U = \lambda^2 \Sigma_{XX} U$$

or $\Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} U = \lambda^2 U$ if Σ_{XX} is also invertible. It is clear we can solve only one eigenvalue problem and then obtain V by (2). Moreover, we can transform this formula to symmetric standard eigenproblem. First way is that letting $(A, B) = (\Sigma_{XX}^{1/2} U, \Sigma_{YY}^{1/2} V)$, then (A, B) can be obatined from singular value decomposition (SVD) of $\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}$ or eigen value decomposition (EVD) of $\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XY} \Sigma_{XX}^{-1/2}$. Another way is using Cholesky decomposition, $\Sigma_{XX} = L_{XX} L_{XX}^\top$. Letting $W = L_{XX}^\top U$, we can rewrite the eigenvalue problem as

$$L_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} L_{XX}^{-\top} W = \lambda^2 W$$

We treat CCA as finding the k-dimensional subsapce in which the projections of x and y are maximally correlated. Therefore we can fomulate as following:

$$\max_{U\in\mathbb{R}^{d_x\times k},V\in\mathbb{R}^{d_y\times k}} Tr(U^\top\Sigma_{XY}V) \quad \text{s.t.} \quad U^\top\Sigma_{XX}U = I = V^\top\Sigma_{YY}V$$

which can be rewritten as a distance minimization problem, i.e.

$$\min_{U\in\mathbb{R}^{d_x\times k},V\in\mathbb{R}^{d_y\times k}} E[||U^\top X - V^\top Y||_2^2] \quad \text{s.t.} \quad U^\top\Sigma_{XX}U = I = V^\top\Sigma_{YY}V$$

In practice, we do not know the distribution of data and will collect i.i.d. draws from this unknown distribution. By replacing the covariance matrix by the sample covariance matrix and maximizing the empirical version of optimization, which is called empirical risk minimization (ERM), we can estimate the canonical correlation coefficients.

3.3 2D CCA

Consider two random matrices, $\mathbf{X}\in\mathbb{R}^{m_x\times n_x}, \mathbf{Y}\in\mathbb{R}^{m_y\times n_y}$. Two-dimensional canonical correlation analysis (2DCCA) [30] seeks left and right tranformations, L_x, R_x, L_y, r_y which maximize correlations between $L_x^\top\mathbf{X}R_x$ and $L_y^\top\mathbf{Y}R_y$. 2DCCA can be fomulated by following:

$$\max_{L_x,R_x,L_y,R_y} \text{cov}(L_x^\top\mathbf{X}R_x, L\text{·}y^\top YR_y) \quad \text{s.t.} \quad \text{var}(L_x^\top\mathbf{X}R_x) = 1 = \text{var}(L_y^\top\mathbf{Y}R_y)$$

Fixing R_x, R_y, XR_x, YR_y are random vectors which allows us to get L_x, L_y using 1DCCA. Then we fix L_x, L_y and solve for R_x, R_y. Continue this procedure until the L_x, R_x, L_y, R_y converge. This is a algorithms provided by [30] and its detail is in algorithm 1. Rather than transforming data to vector and applying 1DCCA, we use 2DCCA for two major benefits: reducing computational complexity and preserving spatial structure of matrix-valued data.

Algorithm 1: iteration method for 2DCCA

Input : $\{\mathbf{X}_t\in\mathbb{R}^{m_x\times n_x}\}_{t=1}^N$, $\{\mathbf{Y}_t\in\mathbb{R}^{m_y\times n_y}\}_{t=1}^N$, k_1, k_2, R_x, R_y

Initialize: R_x, R_x

Center data

repeat

$X_t^r = \mathbf{X}_t R_x, Y_t^r = \mathbf{Y}_t R_x$

$S_{XX}^r = \frac{1}{N}\sum_{t=1}^N X_t^r X_t^{r\top}$, $S_{XY}^r = \frac{1}{N}\sum_{t=1}^N X_t^r Y_t^{r\top}$, $S_{YY}^r = \frac{1}{N}\sum_{t=1}^N Y_t^r Y_t^{r\top}$

compute following k_1 largest generalized eigenvectors to get L_x, L_y

$$\begin{bmatrix} 0 & S_{XY}^r \\ S_{XY}^{r\top} & 0 \end{bmatrix}\begin{bmatrix} L_x \\ L_y \end{bmatrix} = \lambda\begin{bmatrix} S_{XX}^r + R_x I & 0 \\ 0 & S_{YY}^r + R_y I \end{bmatrix}\begin{bmatrix} L_x \\ L_y \end{bmatrix}$$

$X_t^l = \mathbf{X}_t^\top L_x, Y_t^l = \mathbf{Y}_t^\top L_y$

$S_{XX}^l = \frac{1}{N}\sum_{t=1}^N X_t^l X_t^{l\top}$, $S_{XY}^l = \frac{1}{N}\sum_{t=1}^N X_t^r Y_t^{l\top}$, $S_{YY}^l = \frac{1}{N}\sum_{t=1}^N Y_t^l Y_t^{l\top}$

compute following k_1 largest generalized eigenvectors to get R_x, R_y

$$\begin{bmatrix} 0 & S_{XY}^l \\ S_{XY}^{l\top} & 0 \end{bmatrix}\begin{bmatrix} R_x \\ R_y \end{bmatrix} = \lambda\begin{bmatrix} S_{XX}^l + R_x I & 0 \\ 0 & S_{YY}^l + R_y I \end{bmatrix}\begin{bmatrix} R_x \\ R_y \end{bmatrix}$$

until *converged*;

Output : L_x, R_x, L_y, R_y

When we deal with matrix-valued data, the natural way is reshaping them into vectos. Recall 1DCCA,

$$\max_{U,V} \operatorname{corr}(U^\top \operatorname{vec}(X), V^\top \operatorname{vec}(Y))$$

We can rewrite it as

$$\max_{U,V} \operatorname{corr}(Tr(UX), Tr(VY))$$

If U, V are restricted to rank one, then using the fact that rank one matrix can be express to the tensor product of two vectors, we can transform

$$\max_{U,V:\operatorname{rank}(U)=1=\operatorname{rank}(V)} \operatorname{corr}(Tr(UX), Tr(VY))$$

to

$$\max_{L_x,R_x,L_y,R_y} \operatorname{corr}(Tr(R_x L_x^\top X), Tr(R_y L_y^\top Y))$$

which is the case of 2DCCA. Therefore, we can treat 2DCCA as 1DCCA with low ranking restriction and using the trick of matrix factorization to solve the optimization with low rank restriction.

3.4 Convergence Analysis via Lojasiewicz Inequality

Consider an optimization problem:

$$\min_{Z\in\mathbb{R}^p} f(Z) \tag{3}$$

Note that we do not assume f is convex. Our aim is to apply following results to gradient based algorithms for solving (3) and gettint linear or sublinear convergence rate. The key ingredient of this method is the Lojasiewicz gradient inequality as following:

Lemma 3.1 (Lojasiewicz gradient inequality)**.** *Let f be a real analytic function on a neighborhood of X in $\mathbb{R}^n$. Then there are constants $c > 0$ and $\theta \in (0, 1/2]$ such that*

$$|f(Y) - f(X)|^{1-\theta} \le c\|\nabla f(Y)\| \tag{L}$$

where Y in some neighborhood of X.

Since the optimization problem of CCA is Polynomial, we can use Lojasiewicz gradient inequality to our convergence analysis. Considering some iteration Z_k, we make the following assumptions.

- **Primary descent condition:** there exists $\sigma > 0$ such that for large enough k it holds that

$$f(Z_k) - f(Z_{k+1}) \ge \sigma\|\nabla f(Z_k)\|\|Z_k - Z_{k+1}\| \tag{A1}$$

- **Stationary condition:** for large enough k it holds that

$$\nabla f(Z_k) = 0 \quad \Rightarrow \quad Z_k = Z_{k+1} \tag{A2}$$

- **Asymptotic small step-size safeguard:** there exists $\kappa > 0$ such that for large enough k it holds that

$$\|Z_{k+1} - Z_k\| \ge \kappa\|\nabla f(Z_k)\| \tag{A3}$$

Then we have following theorem which is the main tool we use in our analysis.

Theorem 3.2. *Under the condition of lemma 3.1 and assumptions (A1) and (A2), if there exists a cluster point Z^* of the sequence (Z_k) satisfying (L), it is actually its limit, i.e. $Z_k \to Z^*$. Further if (A3) holds, then the convergence rate can be estimated by*

$$\|Z_k - Z^*\| \lesssim \begin{cases} e^{-ck} & \text{if } \theta = \frac{1}{2} \text{ for some } c > 0 \\ k^{-\theta} & \text{if } 0 < \theta < \frac{1}{2} \end{cases}$$

Moreover, $\nabla f(Z_k) \to 0$

See [1, 43] for proofs and more discussions.

4 Tensor Canonical Correlation Analysis

We assume $\mathcal{X}$ and $\mathcal{Y}$ have same mode and shape though it is totally unnescessay in analysis and algorithm. Recall that the CCA aims to find a linear transformation for two multidiemsional variable to maximaize the correlation. Due to this explaination, there is a natural extension to matrix or tensor case. Consider two random tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ and $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ assumed mean zero. Tensor Canonical Correlation Analaysis (TCCA) seeks two rank-one tensors $\mathcal{U} = U_1 \circ \cdots \circ U_m \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ and $\mathcal{V} = V_1 \circ \cdots \circ V_m \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ which maximize correlation between $\langle \mathcal{U}, \mathcal{X} \rangle$ and $\langle \mathcal{V}, \mathcal{Y} \rangle$ and this can be fomulated by

$$\max_{\mathcal{U},\mathcal{V}} \operatorname{Corr}(\langle \mathcal{U}, \mathcal{X} \rangle, \langle \mathcal{V}, \mathcal{Y} \rangle)$$

or equivalently

$$\max_{\mathcal{U},\mathcal{V}} \operatorname{Cov}(\langle \mathcal{U}, \mathcal{X} \rangle, \langle \mathcal{V}, \mathcal{Y} \rangle) \text{ such that } \operatorname{Var}(\langle \mathcal{U}, \mathcal{X} \rangle) = 1 = \operatorname{Var}(\langle \mathcal{V}, \mathcal{Y} \rangle)$$

Since the population distribution is unknown, we can approach this optimization problem by empirical risk minimization or called sample average approximation. That is given samples $\{\mathcal{X}_t, \mathcal{Y}_t\}_t^n$ from the unknown distribution we replace auto-covariance and cross-covariance matrices by empirical estimation, which given us following form:

$$\max_{\mathcal{U},\mathcal{V}} \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{U}, \mathcal{X}_t \rangle \langle \mathcal{V}, \mathcal{Y}_t \rangle \text{ such that } \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{U}, \mathcal{X}_t \rangle^2 = 1 = \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{V}, \mathcal{Y}_t \rangle^2$$

or alternatively

$$\min_{\mathcal{U},\mathcal{V}} \frac{1}{n} \sum_{t=1}^{n} (\langle \mathcal{U}, \mathcal{X}_t \rangle - \langle \mathcal{V}, \mathcal{Y}_t \rangle)^2 \text{ such that } \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{U}, \mathcal{X}_t \rangle^2 = 1 = \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{V}, \mathcal{Y}_t \rangle^2$$

The sample correlation is defined by

$$\rho(\mathcal{U}, \mathcal{V}) = \frac{\frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{U}, \mathcal{X}_t \rangle \langle \mathcal{V}, \mathcal{Y}_t \rangle}{\sqrt{\frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{U}, \mathcal{X}_t \rangle^2 \frac{1}{n} \sum_{t=1}^{n} \langle \mathcal{V}, \mathcal{Y}_t \rangle^2}}$$

Now we define some notations used in algorithm section. An algorithms we consider produce sequence iterates $\{U_{kj}\}$ for every mode $j = 1, \ldots, m$ and we introduce the notation

$$\mathcal{U}_{kj} = U_{k,1} \circ \cdots \circ U_{k,j} \circ U_{k-1,j+1} \cdots \circ U_{k-1,m}$$

For convenience, let $\mathcal{U}_k = \mathcal{U}_{t0} = \mathcal{U}_{k-1,d}$. For a tensor $\mathcal{X}$ and a mode $j = 1, \ldots, m$, the partial contraction X_{kj} is the vector in $\mathbb{R}^{d_j}$ whose i-th entry is

$$(X_{kj})_i = \sum_{i_1=1}^{d_1} \cdots \sum_{i_{j-1}=1}^{d_{j-1}} \sum_{i_{j+1}=1}^{d_{j+1}} \cdots \sum_{i_m=1}^{d_m} (\mathcal{X})_{i_1 \ldots i_{j-1}, i, i_{j+1} \ldots i_x} (U_{k1})_{i_1} (U_{k,j-1})_{i_{j-1}} (U_{k-1,j+1})_{i_{j+1}} (U_{k-1,m})_{i_m}$$

or equivalently,

$$X_{kj} = \mathcal{X} \times_1 U_{k1} \cdots \times_{j-1} U_{k,j-1} \times_{j+1} U_{k-1,j+1} \cdots \times_m U_{k-1,m}$$

Also, let $X_j = \mathcal{X} \times_1 U_1 \cdots \times_{j-1} U_{j-1} \times_{j+1} U_{j+1} \cdots \times_m U_m$. Given data $(\{\mathcal{X}_t, \mathcal{Y}_t\}_{t=1}^n)$, define partial contraction matrix $\mathbf{X}_{kj} \in \mathbb{R}^{n \times d_j}$ whose element is

$$(\mathbf{X}_{kj})_{ti} = \sum_{i_1=1}^{d_1} \cdots \sum_{i_{j-1}=1}^{d_{j-1}} \sum_{i_{j+1}=1}^{d_{j+1}} \cdots \sum_{i_m=1}^{d_m} (\mathcal{X}_t)_{i_1 \ldots i_{j-1}, i, i_{j+1} \ldots i_x} (U_{k1})_{i_1} (U_{k,j-1})_{i_{j-1}} (U_{k-1,j+1})_{i_{j+1}} (U_{k-1,m})_{i_m}$$

or equivalently the t-th row vector of $\mathbf{X}_{kj}$ is

$$\mathcal{X}_t \times_1 U_{k1} \cdots \times_{j-1} U_{k,j-1} \times_{j+1} U_{k-1,j+1} \cdots \times_m U_{k-1,m}$$

4.1 Algorithms

4.1.1 SVD-based algorithm

In the original paper of 2DCCA [30], we allow two modes vary while other modes are fixed. Then this sub-problem will become the 1DCCA case, which can solve it as generalized eigendecomposition. Note that it is unnecessary to pick the same mode in two tensors. However, we only present the version using same mode and fix the order of mode. This algorithm is detailed in Algorithm 2.

Algorithm 2: SVD-based algorithm

Input : $\{\mathcal{X}_t \in \mathbb{R}^{d_1 \times \cdots \times d_m}\}_{t=1}^n$, $\{\mathcal{Y}_t \in \mathbb{R}^{d_1 \times \cdots \times d_m}\}_{t=1}^n$

while *not converged* **do**

  for $j = 1, 2, 3, \ldots m$ **do**

    $\mathbf{S}_{kj}^{XX} = \frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{X}_{kj}$, $\mathbf{S}_{kj}^{XY} = \frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{Y}_{kj} = (\mathbf{S}_{kj}^{YX})^\top$, $\mathbf{S}_{kj}^{YY} = \frac{1}{n}\mathbf{Y}_{kj}^\top \mathbf{Y}_{kj}$

    compute the following largest generalized eigenvectors to get U_{kj}, V_{kj}

$$\begin{bmatrix} 0 & \mathbf{S}_{kj}^{XY} \\ \mathbf{S}_{kj}^{YX} & 0 \end{bmatrix} \begin{bmatrix} U_{kj} \\ V_{kj} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{S}_{kj}^{XX} & 0 \\ 0 & \mathbf{S}_{kj}^{YY} \end{bmatrix} \begin{bmatrix} U_{kj} \\ V_{kj} \end{bmatrix}$$

  end

  $k = k + 1$

end

Output : $\mathcal{U}_k$, $\mathcal{V}_k$

4.1.2 Higher-order Power Method

Recall the MER problem of TCCA

$$\min_{\mathcal{U},\mathcal{V}} \frac{1}{2n}\sum_{t=1}^{n}(\langle \mathcal{U}, \mathcal{X}_t\rangle - \langle \mathcal{V}, \mathcal{Y}_t\rangle)^2 \text{ such that } \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{U}, \mathcal{X}_t\rangle^2 = 1 = \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{V}, \mathcal{Y}_t\rangle^2$$

Using Lagrange multipliers, we have

$$\mathcal{L}(\mathcal{U}, \mathcal{V}, \lambda, \mu) = \frac{1}{2n}\sum_{t=1}^{n}(\langle \mathcal{U}, \mathcal{X}_t\rangle - \langle \mathcal{V}, \mathcal{Y}_t\rangle)^2 + \lambda(1 - \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{U}, \mathcal{X}_t\rangle^2) + \mu(1 - \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{V}, \mathcal{Y}_t\rangle^2) \tag{4}$$

If we only solve one component while others component are fixed, it is the least square problem and $\nabla_{U_j}\mathcal{L} = 0 = \nabla_\lambda \mathcal{L}$ results in solving this two equation

$$\begin{aligned} \frac{1-2\lambda}{n}\mathbf{X}_{kj}^\top \mathbf{X}_{kj} U_j &= \frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_j \\ 1 - 2\lambda &= U_j^\top (\frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{X}_{kj}) U_j \end{aligned} \tag{5}$$

Combining two equations, we have the updating rule

$$U_j = \frac{\mathbf{X}_{kj}^\dagger \mathbf{Y}_{kj} V_j}{\sqrt{V_j^\top \mathbf{Y}_{kj}^\top \mathbf{X}_{kj} \mathbf{X}_{kj}^\dagger \mathbf{Y}_{kj} V_j}}$$

where $\mathbf{X}_{kj}^\dagger$ is the pseudo inverse of $\mathbf{X}_{kj}$. This is the form of power method. Cyclically updating each component yields higher-order power method whose detail is in below.

Algorithm 3: Higher-order Power Method

Input : $\{\mathcal{X}_t\}_{t=1}^n$, $\{\mathcal{Y}_t\}_{t=1}^n$
while *not converged* **do**
  for $j = 1, 2, \ldots, m$ **do**
    Solve the linear system $\mathbf{X}_{kj}^\top \mathbf{X}_{kj} \tilde{U}_{kj} = \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k-1,j}$ to get $\tilde{U}_{kj}$
    $U_{kj} = \tilde{U}_{kj}(\tilde{U}_{kj}^\top (\frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{X}_{kj})\tilde{U}_{kj})^{-1/2}$
    Solve the linear system $\mathbf{Y}_{kj}^\top \mathbf{Y}_{kj} \tilde{U}_{kj} = \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} U_{kj}$ to get $\tilde{V}_{kj}$
    $V_{kj} = \tilde{V}_{kj}(\tilde{V}_{kj}^\top (\frac{1}{n}\mathbf{Y}_{kj}^\top \mathbf{Y}_{kj})\tilde{V}_{kj})^{-1/2}$
  end
  $k = k + 1$
end
Output : $\mathcal{U}_k$, $\mathcal{V}_k$

It is easy to see that SVD-based algorithm is the update rule solving $\mathcal{L}$ with fixing two indexes and instead in HOPM we only fix one index in every iteration.

4.1.3 Alternating Least Squares

In CCA problem, only the projection space matters. However, normalization steps is essential for preventing $\mathcal{U}$ and $\mathcal{V}$ converge to zero quickly. We restrict the components of $\mathcal{U}$ and $\mathcal{V}$ to be of norm

one and call this Alternating Least Square (ALS) to distinguish ALS from HOPM. The detail is following

Algorithm 4: Alternating Least Squares

Input : $\{\mathcal{X}_t\}_{t=1}^n$, $\{\mathcal{Y}_t\}_{t=1}^n$
while *not converged* **do**
  for $j = 1, 2, \ldots, m$ **do**
    Solve the linear system $\mathbf{X}_{kj}^\top \mathbf{X}_{kj} \tilde{U}_{kj} = \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k-1,j}$ to get $\tilde{U}_{kj}$
    $U_{kj} = \tilde{U}_{kj} / \|\tilde{U}_{kj}\|$
    Solve the linear system $\mathbf{Y}_{kj}^\top \mathbf{Y}_{kj} \tilde{U}_{kj} = \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} U_{kj}$ to get $\tilde{V}_{kj}$
    $V_{kj} = \tilde{V}_{kj} / \|\tilde{V}_{kj}\|$
  end
  $k = k + 1$
end
Output : $\mathcal{U}_k$, $\mathcal{V}_k$

We want to construct the connect between ALS and HOPM and give a rigorous analysis of ALS. Consider this modified loss (potential) function by add two extra variables to normalize the components of $\mathcal{U}, \mathcal{V}$

$$\tilde{\mathcal{L}}(\alpha, \mathcal{U}, \beta, \mathcal{V}, \lambda, \mu) = \frac{1}{2n} \sum_{t=1}^n (\langle \alpha \mathcal{U}, \mathcal{X}_t \rangle - \langle \beta \mathcal{V}, \mathcal{Y}_t \rangle)^2 + \lambda (1 - \frac{1}{n} \sum_{t=1}^n \langle \alpha \mathcal{U}, \mathcal{X}_t \rangle^2) + \mu (1 - \frac{1}{n} \sum_{t=1}^n \langle \beta \mathcal{V}, \mathcal{Y}_t \rangle^2) \tag{6}$$

and image that ALS is following this dynamical process:

$$\begin{aligned}
\tilde{U}_{kj} &= \mathbf{X}_{kj}^\dagger \mathbf{Y}_{kj} V_{k,j-1} \\
U_{kj} &= \tilde{U}_{kj} / \|\tilde{U}_{kj}\| \\
\alpha_{kj} &= \left(U_{k,j}^\top \mathbf{X}_{kj}^\top \mathbf{X}_{kj} U_{k,j} \right)^{-1/2} \\
1 - 2\lambda_{kj} &= \alpha_{kj} \beta_{k-1,j} U_{kj}^\top \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k-1,j} = \rho(\mathcal{U}_{kj}, \mathcal{V}_{k,j-1}) \\
\tilde{V}_{kj} &= \mathbf{Y}_{kj}^\dagger \mathbf{X}_{kj} U_{kj} \\
V_{kj} &= \tilde{V}_{kj} / \|\tilde{V}_{kj}\| \\
\beta_{kj} &= \left(V_{k,j}^\top \mathbf{Y}_{kj}^\top \mathbf{Y}_{kj} V_{k,j} \right)^{-1/2} \\
1 - 2\mu_{kj} &= \alpha_{kj} \beta_{kj} U_{kj}^\top \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k,j} = \rho(\mathcal{U}_{kj}, \mathcal{V}_{kj})
\end{aligned} \tag{7}$$

which is motivated by letting

$$\begin{aligned}
\nabla_{\alpha, U_j, \lambda} \tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{kj}, \lambda_{kj}, \beta_{k,j-1}, \mathcal{V}_{k,j-1}, \mu_{k,j-1}) = 0 \\
\nabla_{\beta, V_j, \mu} \tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{kj}, \lambda_{kj}, \beta_{kj}, \mathcal{V}_{kj}, \mu_{kj}) = 0
\end{aligned}$$

where

$$\begin{aligned}\nabla_{U_j}\tilde{\mathcal{L}} &= \frac{\alpha^2(1-2\lambda)}{n}\mathbf{X}_j^\top\mathbf{X}_j U_j - \frac{\alpha\beta}{n}\mathbf{X}_j^\top\mathbf{Y}_j V_j \\ \nabla_{\alpha}\tilde{\mathcal{L}} &= \frac{\alpha(1-2\lambda)}{n}U_j^\top\mathbf{X}_j^\top\mathbf{X}_j U_j - \frac{\beta}{n}U_j^\top\mathbf{X}_j^\top\mathbf{Y}_j V_j \\ \nabla_{\lambda}\tilde{\mathcal{L}} &= 1 - \frac{\alpha^2}{n}U_j^\top\mathbf{X}_j^\top\mathbf{X}_j U_j\end{aligned} \tag{8}$$

Next proposition proves the intuition that HOPM and ALS are equivalent based on the fact that correlation is scalar invariant.

Proposition 4.1. *Let (U_{kj}, V_{kj}), (A_{kj}, B_{kj}) denote the iterates generated by Algorithms 3 and 4 with the same starting guess, respectively. Then it holds*

$$U_{kj} = \frac{A_{kj}}{\sqrt{A_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})A_{kj}}}, V_{kj} = \frac{B_{kj}}{\sqrt{B_{kj}^\top(\frac{1}{n}\mathbf{Y}_{kj}^\top\mathbf{Y}_{kj})B_{kj}}}$$

and

$$\rho(\mathcal{U}_{kj}, \mathcal{V}_{kj}) = \rho(\mathcal{A}_{kj}, \mathcal{B}_{kj})$$

Moreover, if $(\alpha, A_1, \ldots, A_m, \lambda, \beta, B_1, \ldots, B_m, \mu)$ is a critical point of the modified loss $\tilde{\mathcal{L}}$and $\alpha, \beta > 0$, then $(\alpha^{(1/m)}A_1, \ldots, \alpha^{(1/m)}A_m, \beta^{(1/m)}, \lambda, B_1, \ldots, \beta^{(1/m)}B_m, \mu)$ is a critical point of the original loss $\mathcal{L}$

Proof. It suffices to show that there exists $a_{kj} > 0, b_{kj} > 0$ for all k, j such that

$$U_{kj} = a_{kj}A_{kj}, V_{kj} = b_{kj}B_{kj}$$

We do this by induction. Since they have same stating guess, this hold for $k = 0, j = 0$. By hypothesis and construction of A_{kj} we have

$$\begin{aligned}U_{kj} &= \frac{\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}}{\sqrt{V_{k,j-1}^\top\mathbf{Y}_{kj}^\top\mathbf{X}_{kj}\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}}} \\ &= \frac{b_{kj}\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}B_{k,j-1}}{\sqrt{V_{k,j-1}^\top\mathbf{Y}_{kj}^\top\mathbf{X}_{kj}\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}}} \\ &= \frac{b_{kj}\|\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}B_{k,j-1}\|A_{kj}}{\sqrt{V_{k,j-1}^\top\mathbf{Y}_{kj}^\top\mathbf{X}_{kj}\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}}}\end{aligned}$$

We can do the the same argument for V_{kj}. Because all constants are positive and correlation is scalar invariant, this yields the desired result. Moreover, by construction, we have

$$1 = U_{kj}^\top\mathbf{X}_{kj}^\top\mathbf{X}_{kj}U_{kj} = a_{kj}^2A_{kj}^\top\mathbf{X}_{kj}^\top\mathbf{X}_{kj}A_{kj}$$

Following same argument for $V_{kj} = B_{kj}/\sqrt{B_{kj}^\top(\frac{1}{n}\mathbf{Y}_{kj}^\top\mathbf{Y}_{kj})B_{kj}}$, we complete the first part of proposition. For second part, we have

$$\begin{aligned}
&\alpha^{(1/m)}\nabla_{U_j}\mathcal{L}(\alpha^{(1/m)}A_1,\ldots,\alpha^{(1/m)}A_m,\beta^{(1/m)}\lambda,B_1,\ldots,\beta^{(1/m)}B_m,\mu)\\
=&\frac{\alpha^2(1-2\lambda)}{n}\mathbf{X}_j^\top\mathbf{X}_jA_j-\frac{\alpha\beta}{n}\mathbf{X}_j^\top\mathbf{Y}_jB_j\\
=&\nabla_{U_j}\tilde{\mathcal{L}}(\alpha,A_1,\ldots,A_m,\lambda,\beta,B_1,\ldots,B_m,\mu)\\
=&0
\end{aligned}$$

and

$$\begin{aligned}
&\nabla_{\lambda}\mathcal{L}(\alpha^{(1/m)}A_1,\ldots,\alpha^{(1/m)}A_m,\beta^{(1/m)}\lambda,B_1,\ldots,\beta^{(1/m)}B_m,\mu)\\
=&1-\frac{\alpha^2(1-2\lambda)}{n}\mathbf{X}_j^\top\mathbf{X}_jA_j\\
=&\nabla_{\lambda}\tilde{\mathcal{L}}(\alpha,A_1,\ldots,A_m,\lambda,\beta,B_1,\ldots,B_m,\mu)\\
=&0
\end{aligned}$$

Since $\alpha^{(1/m)}>0$, $\nabla_{U_j}\mathcal{L}(\alpha^{(1/m)}A_1,\ldots,\alpha^{(1/m)}A_m,\beta^{(1/m)}\lambda,B_1,\ldots,\beta^{(1/m)}B_m,\mu)=0$. By doing the same argument for $\nabla_\beta\mathcal{L}$ and $\nabla_\mu\mathcal{L}$, the theorem follows. □

4.2 Convergence Analysis

Recall the updating rule of ALS:

$$\begin{aligned}
\tilde{U}_{kj}&=\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}\\
U_{kj}&=\tilde{U}_{kj}/\|\tilde{U}_{kj}\|\\
\alpha_{kj}&=\left(U_{k,j}^\top\mathbf{X}_{kj}^\top\mathbf{X}_{kj}U_{k,j}\right)^{-1/2}\\
1-2\lambda_{kj}&=\alpha_{kj}\beta_{k-1,j}U_{kj}^\top\mathbf{X}_{kj}^\top\mathbf{Y}_{kj}V_{k-1,j}=\rho(\mathcal{U}_{kj},\mathcal{V}_{k,j-1})\\
\tilde{V}_{kj}&=\mathbf{Y}_{kj}^\dagger\mathbf{X}_{kj}U_{kj}\\
V_{kj}&=\tilde{V}_{kj}/\|\tilde{V}_{kj}\|\\
\beta_{kj}&=\left(V_{k,j}^\top\mathbf{Y}_{kj}^\top\mathbf{Y}_{kj}V_{k,j}\right)^{-1/2}\\
1-2\mu_{kj}&=\alpha_{kj}\beta_{kj}U_{kj}^\top\mathbf{X}_{kj}^\top\mathbf{Y}_{kj}V_{k,j}=\rho(\mathcal{U}_{kj},\mathcal{V}_{kj})
\end{aligned}\tag{9}$$

Assumption 1. *We make three assumptions:*

- $0<\sigma_{l,x}=:\sigma_{\min}(\frac{1}{n}\sum_{t=1}^n vec(\mathcal{X}_t)vec(\mathcal{X}_t)^\top)<\sigma_{\max}(\frac{1}{n}\sum_{t=1}^n vec(\mathcal{X}_t)vec(\mathcal{X}_t)^\top):=\sigma_{u,x}<\infty$
- $0<\sigma_{l,y}=:\sigma_{\min}(\frac{1}{n}\sum_{t=1}^n vec(\mathcal{Y}_t)vec(\mathcal{Y}_t)^\top)<\sigma_{\max}(\frac{1}{n}\sum_{t=1}^n vec(\mathcal{Y}_t)vec(\mathcal{Y}_t)^\top):=\sigma_{u,y}<\infty$
- $\rho(\mathcal{U}_0,\mathcal{V}_0)>0$

Note that the third condition always can be true since if $\rho(\mathcal{U}_0,\mathcal{V}_0)<0$ we can flip the sign of component of $\mathcal{U}_0$ or $\mathcal{V}_0$ such that $\rho(\mathcal{U}_0,\mathcal{V}_0)>0$

Theorem 4.2. *If assumption 1 holds, then the dynamic 9 satisfies condition (A1), (A2) and (A3). Thus, the iterates U_{kj}, V_{kj} generated by ALS converge to a stationary point and the convergence rate depends on the exponent in the Lojasiewicz gradient inequality.*

Proof. It is clear that $\tilde{\mathcal{L}}$ is analytic, so it suffices to verify three conditions in Theorem 3.2.

(Stationary Condition) Since other variables only depend on $U_{k+1,j}$ and $V_{k+1,j}$, it suffices to show that $U_{kj} = U_{k+1,j}$ and $V_{kj} = V_{k+1,j}$. By symmetric argument, we only show the part for $U_{k+1,j}$. Note that $\nabla_{U_j}\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j}) = 0$ implies $\alpha_{kj}(1-\lambda_{kj})\mathbf{X}_{kj}^\top\mathbf{X}_{kj}U_{kj} = \mathbf{X}_{kj}^\top\mathbf{Y}_{kj}V_{k-1,j}$, so we have $\tilde{U}_{k+1,j} = \alpha_{kj}(1-\lambda_{kj})U_{kj}$. After normalization, we get $U_{k,j} = U_{k+1,j}$, and the stationary condition follows.

(Asymptotic small step-size safeguard) Again, it suffices to show the part of U_{kj} by the same argument. We first show the following lemma.

Lemma 4.3. *Under assumption 1, we have for all j, k*

1. $U_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})U_{kj} \in [\sigma_{l,x}, \sigma_{u,x}]$
2. $\alpha_{kj} \in [\sigma_{u,x}^{-1/2}, \sigma_{l,x}^{-1/2}]$

Proof. We only prove the case $m = 2$ but it is trivial to extend to arbitrary m. For the first statement, it follows by this two identities that

$$U_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})U_{kj} = (U_{k1}\otimes U_{k2})^\top(\frac{1}{n}\sum_{t=1}^{n}\mathrm{vec}(\mathcal{X}_t)\mathrm{vec}(\mathcal{X}_t)^\top)(U_{k1}\otimes U_{k2})$$

and

$$(U_{k1}\otimes U_{k2})^\top(U_{k1}\otimes U_{k2}) = (U_{k1}^\top U_{k1}\otimes U_{k2}^\top U_{k2}) = \|U_{k1}\|^2\|U_{k2}\|^2 = 1$$

The second statement just follows by the definition of α and the statement 1. □

Moreover, since $(1-2\lambda_{k,j}) = \rho(\mathcal{U}_{kj}, \mathcal{V}_{kj})$ is bounded, $(\alpha_{kj}, U_{kj}\lambda_{kj})$ is on compact set. Combining this fact and

$$\nabla_{\alpha,U_j,\lambda}\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) = 0$$
$$\nabla_{\beta,V_j,\mu}\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j}, \mathcal{V}_{k+1,j}, \mu_{k+1,j}) = 0$$

we deduce for some $L > 0$ independent on k, j

$$\begin{aligned}
&\|\nabla\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0})\|^2 \\
=&\sum_j \|\nabla_{U_j}\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0}) - \nabla_{U_j}\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1})\|^2 \\
+&\|\nabla_{\alpha,\lambda}\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0}) - \nabla_{\alpha,\lambda}\tilde{\mathcal{L}}(\alpha_{km}, \mathcal{U}_{km}, \lambda_{km}, \beta_{k,m-1}, \mathcal{V}_{k,m-1}, \mu_{k,m-1})\|^2 \\
+&\sum_j \|\nabla_{V_j}\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0}) - \nabla_{V_j}\nabla\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j}, \mathcal{V}_{k+1,j}, \mu_{k+1,j})\|^2 \\
+&\|\nabla_{\beta,\mu}\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0}) - \nabla_{\beta,\mu}\tilde{\mathcal{L}}(\alpha_{km}, \mathcal{U}_{km}, \lambda_{km}, \beta_{k,m}, \mathcal{V}_{k,m}, \mu_{k,m})\|^2 \\
\leq&L^2(2m+4)\|(\alpha_{k0}, \mathcal{U}_{k0}, \lambda_{k0}, \beta_{k0}, \mathcal{V}_{k0}, \mu_{k0}) - (\alpha_{km}, \mathcal{U}_{km}, \lambda_{km}, \beta_{k,m}, \mathcal{V}_{k,m}, \mu_{k,m}\|^2
\end{aligned}$$

This complete asymptotic small step-size safeguard condition.

(Primary descent condition) We use the fact that updating each component is a least square problem to prove the following lemma.

Lemma 4.4. *Under assumption 1, we have*

1. $1 > \lambda_{k+1,j} - \lambda_{k,j} = [(1 - 2\lambda_{k,j}) - (1 - 2\lambda_{k+1,j})]/2 > 0$ *and, thus,* $\rho(\mathcal{U}_{kj}, \mathcal{V}_{kj})$ *converges.*
2. *there exists* $\sigma_o > 0$ *which is independent on* k, j *such that*

$$\begin{aligned}
&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
-&\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
\geq&\frac{\sigma_0}{2}\left[(\alpha_{k+1,j} - \alpha_{k+1,j+1})^2 + \|U_{kj} - U_{k+1,j}\|^2 + (\lambda_{k+1,j} - \lambda_{k+1,j+1})^2\right]
\end{aligned}$$

Proof. With others variables fixed, it is clear that the problem reduces to 1DCCA problem and $\{\alpha_{k+1,j}, U_{k+1,j}\}$ is the solution. Therefore, the correlation $\rho(\mathcal{U}_{kj}, \mathcal{V}_{kj})$ is increasing at every updating and the first statement follows by assumption.

Note that $(\alpha_{kj}, \mathcal{U}_{kj})$ is feasible, i.e.

$$\frac{\alpha_{kj}^2}{n} U_{kj}^\top \mathbf{X}_{k,j}^\top \mathbf{X}_{k,j} U_{kj} = 1 = \frac{\alpha_{k+1,j}^2}{n} U_{k+1,j}^\top \mathbf{X}_{kj}^\top \mathbf{X}_{kj} U_{k+1,j}$$

so

$$\begin{aligned}
&\frac{1}{2}\Big[\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
&- \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1})\Big] \\
=&\frac{1}{2}\left[\frac{\alpha_{kj}\beta_{k,j-1}}{n} U_{kj}^\top \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k,j-1} - \frac{\alpha_{k+1,j}\beta_{k,j-1}}{n} U_{k+1,j}^\top \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k,j-1}\right] \qquad (10) \\
=&\frac{1}{2}\left[(1 - 2\lambda_{kj}) - (1 - 2\lambda_{k+1,j})\right] \\
\geq&\lambda_{k+1,j} - \lambda_{kj} \\
\geq&(\lambda_{k+1,j} - \lambda_{kj})^2
\end{aligned}$$

where the last inequality is because $1 > \lambda_{k+1,j} - \lambda_{kj} > 0$. Furthermore, we have

$$\begin{aligned}
&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) - \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
=&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) - \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
=&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) - \tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
+&\tilde{\mathcal{L}}(\alpha_{k,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) - \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \qquad (11)
\end{aligned}$$

Due to the statement 1 we know

$$\begin{aligned}
&\nabla_\alpha \tilde{\mathcal{L}}(\alpha_{k+1}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) = 0 \\
&\nabla_\alpha^2 \tilde{\mathcal{L}}(\alpha_{k+1}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) = (1 - 2\lambda_{k+1,j}) U_{k+1,j}^\top (\frac{1}{n}\mathbf{X}_{kj}^\top \mathbf{X}_{kj}) U_{k+1,j} > 0
\end{aligned}$$

which implies

$$\begin{aligned}
&\tilde{\mathcal{L}}(\alpha_{k,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
-&\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
\geq&(1-2\lambda_{0,0})\sigma_{l,x}(\alpha_{k,j}-\alpha_{k+1,j})^2
\end{aligned} \tag{12}$$

Also,

$$\begin{aligned}
&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j-1}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
-&\tilde{\mathcal{L}}(\alpha_{kj}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j-1}, \mathcal{V}_{k+1,j-1}, \mu_{k+1,j-1}) \\
=&\alpha_{k,j}^2(1-2\lambda_{k+1,j})(U_{k+1,j-1}-U_{k+1,j})^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})(U_{k+1,j-1}-U_{k+1,j}) \\
-&\alpha_{kj}\beta_{k,j-1}U_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k,j-1}+\alpha_{kj}\beta_{k,j-1}U_{k+1,j}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k+1,j-1} \\
\geq&\sigma_{u,x}^{-1}(1-2\lambda_{0,0})(U_{k+1,j-1}-U_{k+1,j})^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})(U_{k+1,j-1}-U_{k+1,j})
\end{aligned} \tag{13}$$

where last inequality follows by the fact $U_{k+1,j}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k,j-1} - U_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k,j-1} > 0$ which can be shown by following: Let $f(U) = U^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k,j-1}$ be a linear function w.r.t. U with gradient $(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{Y}_{kj})V_{k,j-1}$. Since $\tilde{U}_{k+1,j} = \mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1}$ and $V_{k,j-1}^\top(\frac{1}{n}\mathbf{Y}_{kj}^\top\mathbf{X}_{kj})\mathbf{X}_{kj}^\dagger\mathbf{Y}_{kj}V_{k,j-1} > 0$ as well as $\|U_{kj}\| = 1 = \|U_{k+1,j}\|$, by the property of project gradient descent on the unit ball, $f((U_{k,j}+\epsilon U_{k+1,j}/\|U_{k,j}+\epsilon U_{k+1,j}\|)) > f(U_{k,j})$ for all $\epsilon > 0$. Letting $\epsilon \to \infty$, we obtain the desired result.

Combining (10), (12), (13), statement 1 and assumption 1, we complete the lemma. □

Following the lemma, we deduce

$$\begin{aligned}
&\tilde{\mathcal{L}}(\alpha_{k0}, \mathcal{U}_{k+1,0}, \lambda_{k,0}, \beta_{k+1,0}, \mathcal{V}_{k+1,0}, \mu_{k+1,0}) - \tilde{\mathcal{L}}(\alpha_{k+1,m}, \mathcal{U}_{k+1,m}, \lambda_{k+1,m}, \beta_{k+1,m}, \mathcal{V}_{k+1,m}, \mu_{k+1,m}) \\
\geq&\sum_{j=1}^{m}\Big[\tilde{\mathcal{L}}(\alpha_{k,j-1}, \mathcal{U}_{k+1,j-1}, \lambda_{k,j-1}, \beta_{k,j}, \mathcal{V}_{k+1,j-1}, \mu_{k,j}) - \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k,j}, \mathcal{V}_{k+1,j-1}, \mu_{k,j}) \\
&+\tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k,j}, \mathcal{V}_{k+1,j-1}, \mu_{k,j}) - \tilde{\mathcal{L}}(\alpha_{k+1,j}, \mathcal{U}_{k+1,j}, \lambda_{k+1,j}, \beta_{k+1,j}, \mathcal{V}_{k+1,j}, \mu_{k+1,j})\Big] \\
\geq&\frac{\sigma_0}{2}\Bigg[\sum_{j=1}^{m}(\alpha_{k,j}-\alpha_{k+1,j})^2+(\lambda_{k,j}-\lambda_{k+1,j})^2+\|U_{k+1,j}-U_{kj}\|^2 \\
&+\sum_{j=1}^{m}(\beta_{k,j}-\beta_{k+1,j})^2+(\mu_{k,j}-\mu_{k+1,j})^2+\|V_{k+1,j}-V_{kj}\|^2\Bigg] \\
\geq&\frac{\sigma_0}{2}\left[(\alpha_{k,0}-\alpha_{k+1,m})^2+(\lambda_{k,0}-\lambda_{k+1,m})^2+\|\mathcal{U}_{k+1,0}-\mathcal{U}_{k+1,m}\|^2\right. \\
&\left.+(\beta_{k,0}-\beta_{k+1,m})^2+(\mu_{k,0}-\mu_{k+1,m})^2+\|\mathcal{V}_{k+1,j}-\mathcal{V}_{kj}\|^2\right]
\end{aligned}$$

where last inequality is due to triangle inequality. It is clear that combining this and asymptotic small step-size safeguard yields the primary descent condition, so the proof is completed. □

4.3 Efficient Algorithms for Large-scale Data

Since solving $\mathbf{X}_{kj}^\top \mathbf{X}_{kj} \tilde{U}_{kj} = \mathbf{X}_{kj}^\top \mathbf{Y}_{kj} V_{k-1,j}$ is equivalent to solving the following least squares problem,

$$\min_{\tilde{U}} f_{kj}(\tilde{U}) := \min_{\tilde{U}} \frac{1}{2n} \|\mathbf{X}_{kj}\tilde{U} - \mathbf{Y}_{kj} V_{k,j-1}\|^2$$

$$\min_{\tilde{V}} g_{kj}(\tilde{V}) := \min_{\tilde{V}} \frac{1}{2n} \|\mathbf{X}_{kj}\tilde{U}_{kj} - \mathbf{Y}_{kj}\tilde{V}\|^2$$

It is convenient to apply advanced optimization methods to this least square problem. The detail is summarized below:

Algorithm 5: Inexact Alternating Least Squares

Input : $\{\mathcal{X}_t\}_{t=1}^n$, $\{\mathcal{Y}_t\}_{t=1}^n$, $\gamma_{x,j}$, $\gamma_{y,j}$
while *not converged* **do**
  for $j = 1, 2, \ldots, m$ **do**
    With initialization $\tilde{U}_{k-1,j}$, solve the following problem up to constant ϵ to get $\tilde{U}_{kj}$:

$$f_{kj}(\tilde{U}_{kj}) \le \min_{\tilde{U}} f_{kj}(\tilde{U}) + \epsilon = \min_{\tilde{U}} \frac{1}{2n} \|\mathbf{X}_{kj}\tilde{U} - \mathbf{Y}_{kj} V_{k,j-1}\|^2 + \frac{\gamma_{x,j}}{2}\|\tilde{U}\|^2 + \epsilon$$

    $U_{kj} = \tilde{U}_j / \|\tilde{U}_{kj}\|$
    With initialization $\tilde{V}_{k-1,j}$, solve the following problem up to constant ϵ to get $\tilde{V}_{kj}$:

$$g_{kj}(\tilde{V}_{kj}) \le \min_{\tilde{V}} g_{kj}(\tilde{V}) + \epsilon = \min_{\tilde{V}} \frac{1}{2n} \|\mathbf{X}_{kj}\tilde{U_{kj}} - \mathbf{Y}_{kj}\tilde{V}\|^2 + \frac{\gamma_{y,j}}{2}\|\tilde{V}\|^2 + \epsilon$$

    $V_{kj} = \tilde{V}_j / \|\tilde{V}_{kj}\|$
  end
  $k = k + 1$
end
Output : $\mathcal{U}_k$, $\mathcal{V}_k$

There are several techniques to improve the convergence:

1. Warm-start: After several loop, f_{kj} varies slightly as U_{ki} for $i \neq j$ and V_{ki} for all i. We may use $\tilde{U}_{kj}$ as initialization for minimizing $f_{k+1,j}(\tilde{U})$.
2. Canonical ridge: ℓ_2 regularization may be used to make the least square problem guaranteed to be strongly convex.

Recall the assumptions we make previous section:

1. $0 < \sigma_{l,x} =: \sigma_{\min}(\frac{1}{n}\sum_{t=1}^n \mathrm{vec}(\mathcal{X}_t)\mathrm{vec}(\mathcal{X}_t)^\top) < \sigma_{\max}(\frac{1}{n}\sum_{t=1}^n \mathrm{vec}(\mathcal{X}_t)\mathrm{vec}(\mathcal{X}_t)^\top) := \sigma_{u,x} < \infty$
2. $0 < \sigma_{l,y} =: \sigma_{\min}(\frac{1}{n}\sum_{t=1}^n \mathrm{vec}(\mathcal{Y}_t)\mathrm{vec}(\mathcal{Y}_t)^\top) < \sigma_{\max}(\frac{1}{n}\sum_{t=1}^n \mathrm{vec}(\mathcal{Y}_t)\mathrm{vec}(\mathcal{Y}_t)^\top) := \sigma_{u,y} < \infty$

Now we show the error bound for the inexact updating.

Theorem 4.5. *Denoting $\{U_{kj}^*, V_{kj}^*\}$ generated by algorithm 4 and U_{kj}, V_{kj} generated by algorithm 5, we have*

$$\max\{\|U_{kj} - U_{kj}^*\|, \|V_{kj} - V_{kj}^*\|\} = O(r^{2mk+j}\sqrt{\epsilon})$$

for some r that depends on $m, \sigma_{u,x}, \sigma_{l,x}, \sigma_{u,y}, \sigma_{l,y}$

Proof. We only focus on the U_{kj} since the similar argument can directly apply to V_{kj}. First, we prove the following lemma which also reveals the fact that the updating variables never go to zero.

Lemma 4.6. *For all k, j, we have*

$$\|U_{kj}\| > \sigma_{u,x}^{-1}\sigma_{l,x}^{1/2}\sigma_{l,y}^{1/2}$$

and

$$\|V_{kj}\| > \sigma_{u,y}^{-1}\sigma_{l,y}^{1/2}\sigma_{l,x}^{1/2}$$

Proof. We only show the first statement. It is easy to see that

$$\begin{aligned}\|U_{kj}\| &= \|(\mathbf{X}_{kj}^\top \mathbf{X}_{kj})^{-1}\mathbf{X}_{kj}\mathbf{Y}_{kj}V_{k,j-1}\| \\ &\geq \sigma_{u,x}\sigma_{l,x}^{1/2}\sigma_{l,y}^{1/2}\end{aligned}$$

where last inequality is due to the fact that for any unit vector U we have

$$U^\top \mathbf{X}_{kj}^\top \mathbf{X}_{kj} U = (U_{k1} \otimes \cdots \otimes U^\top \otimes \cdots \otimes U_{km})^\top \frac{1}{n}\sum_{t=1}^{n} \mathrm{vec}(\mathcal{X}_t)\mathrm{vec}(\mathcal{X}_t)^\top (U_{k1} \otimes \cdots \otimes U^\top \otimes \cdots \otimes U_{km})$$

This complete the proof. □

In this proof, we distinguish the iterates of inexact updating power iterations (Algorithm 5) from the iterates of the exact power iterations (Algorithm 4) and denote the latter with asterisks, i.e., U_{kj} and U_{kj}^*. We denote the exact optimum of $f_{kj}(U)$ and $g_{kj}(V)$ by $\tilde{U}_{kj}^\natural$ and $\tilde{V}_{kj}^\natural$ respectivel, and use tilde to indicate the iterates is unnormalized, i.e., $\tilde{U}_{kj}$ and $\tilde{U}_{kj}^*$.

We prove this theorem by induction. We will show the recurrent relationship of the error bound. By triangle inequality, we have

$$\|U_{kj} - U_{kj}^*\| \leq \|U_{kj} - U_{kj}^\natural\| + \|U_{kj} - U_{kj}^*\|$$

For the first term, by construction, we have

$$\epsilon \geq f_{kj}(\tilde{U}_{kj}) - f_{kj}(\tilde{U}_{kj}^\natural) = \frac{1}{2}(\tilde{U}_{kj} - \tilde{U}_{kj}^\natural)^\top \mathbf{X}_{kj}^\top \mathbf{X}_{kj}(\tilde{U}_{kj} - \tilde{U}_{kj}^\natural) \geq \sigma_{l,x}\|\tilde{U}_{kj} - \tilde{U}_{kj}^\natural\|^2$$

The fact that $\|\tilde{U}_{kj}^\natural|$ is uniformly bounded blow, proved by previous lemma, yields for some $c > 0$

$$\|U_{kj} - U_{kj}^\natural\| \leq \tan^{-1}\left(\frac{\|\tilde{U}_{kj} - \tilde{U}_{kj}^\natural\|}{\|\tilde{U}_{kj}^\natural\|}\right) \leq c\sigma_{l,x}^{-1}\sqrt{\epsilon}$$

For the second term, by construction,

$$\begin{aligned}\|U_{kj} - U^*_{kj}\| &= \|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V_{k,j-1} - ((\mathbf{X}^*_{k,j})^\top \mathbf{X}^*_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ &\le \|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V^*_{k,j-1}\| \\ &+ \|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V^*_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ &+ \|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ &+ \|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1} - ((\mathbf{X}^*_{k,j})^\top \mathbf{X}^*_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\|\end{aligned}$$

where $\mathbf{X}^*_{k,j}$ is the exact version of $\mathbf{X}_{k,j}$. By the same technique we use again and again, we get

$$\begin{aligned}&\|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V^*_{k,j-1}\| \\ \le&\|\frac{1}{n}(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\|\|\frac{1}{\sqrt{n}}\mathbf{X}_{k,j}^\top\|\|\frac{1}{\sqrt{n}}\mathbf{Y}_{k,j}\|\|V^*_{k,j-1} - V^*_{k,j-1}\| \\ \le&\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\|\|V^*_{k,j-1} - V^*_{k,j-1}\|\end{aligned}$$

and

$$\begin{aligned}&\|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}_{k,j} V^*_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ \le&\|\frac{1}{n}(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\|\|\frac{1}{\sqrt{n}}\mathbf{X}_{k,j}^\top\|\|\frac{1}{\sqrt{n}}(\mathbf{Y}_{k,j} - \mathbf{Y}^*_{k,j})\| \\ \le&\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\left(\sum_{i<j}\|V_{k,i} - V^*_{ki}\| + \sum_{i>j}\|V_{k-1,i} - V^*_{k-1,i}\|\right)\end{aligned}$$

and, by similar argument,

$$\begin{aligned}&\|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}\mathbf{X}_{k,j}^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1} - (\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ \le&\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\left(\sum_{i<j}\|U_{k,i} - U^*_{ki}\| + \sum_{i>j}\|U_{k-1,i} - U^*_{k-1,i}\|\right) \\ &\|(\mathbf{X}_{k,j}^\top \mathbf{X}_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1} - ((\mathbf{X}^*_{k,j})^\top \mathbf{X}^*_{k,j})^{-1}(\mathbf{X}^*_{k,j})^\top \mathbf{Y}^*_{k,j} V^*_{k,j-1}\| \\ \le&\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\left(\sum_{i<j}\|U_{k,i} - U^*_{ki}\| + \sum_{i>j}\|U_{k-1,i} - U^*_{k-1,i}\|\right)\end{aligned}$$

In sum, by induction hypothesis, we have for some $c > 0$ (the constant change line by line)

$$\begin{aligned}\|U_{kj} - U^*_{kj}\| \le& c\sigma_{l,x}^{-1}\sqrt{\epsilon} + 2\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\left(\sum_{i<j}\|U_{k,i} - U^*_{ki}\| + \sum_{i>j}\|U_{k-1,i} - U^*_{k-1,i}\|\right. \\ &\left. + \sum_{i<j}\|V_{k,i} - V^*_{ki}\| + \sum_{i>j}\|V_{k-1,i} - V^*_{k-1,i}\|\right)\end{aligned}$$

Similarly, we have

$$\|V_{kj} - V_{kj}^*\| \leq c\sigma_{l,y}^{-1}\sqrt{\epsilon} + 2\sigma_{l,y}^{-1}\sigma_{u,y}^{1/2}\sigma_{u,x}^{1/2}\left(\sum_{i<j}\|V_{k,i} - V_{ki}^*\| + \sum_{i>j}\|V_{k-1,i} - V_{k-1,i}^*\|\right.$$
$$\left. + \sum_{i<j}\|U_{k,i} - U_{ki}^*\| + \sum_{i>j}\|U_{k-1,i} - U_{k-1,i}^*\|\right)$$

Define $c_1 = \max\{c\sigma_{l,y}^{-1}, c\sigma_{l,x}^{-1}\}$ and $c_2 = \max\{2\sigma_{l,y}^{-1}\sigma_{u,y}^{1/2}\sigma_{u,x}^{1/2}, 2\sigma_{l,x}^{-1}\sigma_{u,x}^{1/2}\sigma_{u,y}^{1/2}\}$ and

$$E_\ell = \begin{cases} 0 & \text{if } \ell \leq 0 \\ \|U_{kj} - U_{kj}^*\| & \text{if } \ell > 0 \text{ is odd and } \frac{\ell+1}{2} \bmod 2 = j \\ \|V_{kj} - V_{kj}^*\| & \text{if } \ell > 0 \text{ is even and } \frac{\ell}{2} \bmod 2 = j \end{cases}$$

and $2m$-th generalized Fibonacci number

$$F_\ell = \begin{cases} 0 & \text{if } \ell \leq 1 \\ c_1\sqrt{\epsilon} & \text{if } \ell = 1 \\ c_2(F_{\ell-1} + F_{\ell-2} + \cdots + F_{\ell-2m}) & \text{if if } \ell > 1 \end{cases}$$

then we have

$$\begin{aligned} E_\ell &\leq c_1\sqrt{\epsilon} + c_2(E_{\ell-1} + E_{\ell-2} + \cdots + E_{\ell-2m}) \\ &= (\ell-1)c_1\sqrt{\epsilon} + F_\ell \end{aligned} \tag{14}$$

Following the technique of [27], letting

$$R = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ c_2 & c_2 & c_2 & & c_2 & c_2 \end{bmatrix}$$

we have

$$\begin{bmatrix} F_\ell \\ \vdots \\ F_{\ell+2m-1} \\ F_{\ell+2m} \end{bmatrix} = R^\ell \begin{bmatrix} 0 \\ \vdots \\ 0 \\ c_1\sqrt{\epsilon} \end{bmatrix}$$

Provided that there are $2m$ distinct eigenvalues of R, denoted $r_1, r_2, \ldots, r_{2m}$, which can be shown in different proofs [16, 54], via eigendecomposition $R = VDV^{-1}$ where

$$V = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ r_1 & r_2 & r_3 & & r_{2m} \\ r_1^2 & r_2^2 & r_3^2 & \cdots & r_{2m}^2 \\ & & \vdots & & \\ r_1^{2m-1} & r_2^{2m-1} & r_3^{2m-1} & \cdots & r_{2m}^{2m-1} \end{bmatrix}$$

we have

$$
\begin{aligned}
F_\ell &= [1,0,\ldots,0]VD^\ell V^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ c_1\sqrt{\epsilon} \end{bmatrix} \\
&= [r_1^\ell, r_2^\ell, \ldots, r_{2m}^\ell]V^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ c_1 \end{bmatrix}\sqrt{\epsilon} \\
&= \sum_{i=1}^{2m} r_i^\ell z_i \sqrt{\epsilon}
\end{aligned}
\tag{15}
$$

where $z_1, \ldots, z_{2m}$ satisfy

$$
V\begin{bmatrix} z_1 \\ \vdots \\ z_{2m-1} \\ z_{2m} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ c_1 \end{bmatrix}
$$

Combining (14) and (15), we have

$$
E_\ell = \left[(\ell-1)c_1 + \sum_{i=1}^{2m} r_i^\ell z_i\right]\sqrt{\epsilon} = O(r^\ell\sqrt{\epsilon})
$$

where r is the largest eigenvalue of R, which completes the proof. □

Note that we get the same order for error bound of inexact updating in [51] when $m = 1$.

4.4 Effective Initialization

In this section, we assume $m = 2$ and go back 2DCCA case.

Since the 2DCCA problem is non-convex, there is no guarantee that ALS converges to global maxima, and choosing an initial point is important. Now we demonstrate a way setting up the initial point via 1DCCA. If (C_x, C_y) is the solution of maximizing $\mathrm{corr}(\mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{Y}))$, we use best rank-1 approximation as initialization, that is $U_1 \otimes U_2 \approx C_x$ and $V_1 \otimes V_2 \approx C_y$ which can be obtained by SVD of $\mathrm{unvec}(C_x)$ and $\mathrm{unvec}(C_y)$. Heuristically, initial point using best rank one approximation may result in higher correlation than a random guess, so it is more probable to be close to global maxima. Now we show that in some case, 1DCCA solution is indeed close to 2DCCA solution so rank-1 approximation must lie near the global maximum as sample size increasing.

Consider this model:

$$
\begin{aligned}
X &= c(X_1 \otimes X_2) + \epsilon_x \\
Y &= c(Y_1 \otimes Y_2) + \epsilon_y
\end{aligned}
\tag{16}
$$

where X_1, X_2, Y_1, Y_2 are unit vector with same length and $c, \epsilon_x, \epsilon_y$ are random variables and independent with zero mean and unit variance. Then, we have

$$\begin{aligned}\Sigma_{XX} &= X_1X_1^\top \otimes X_2X_2^\top + I \\ \Sigma_{YY} &= Y_1Y_1^\top \otimes Y_2Y_2^\top + I \\ \Sigma_{XY} &= X_1Y_1^\top \otimes X_2Y_2^\top\end{aligned} \tag{17}$$

Recall the 1DCCA formula,

$$\Sigma_{XX}^{-1}\Sigma_{YX}\Sigma_{YY}^{-1}\Sigma_{XY}C_x = \rho C_x$$

and combine all together

$$\begin{aligned}\rho C_x &= (X_1X_1^\top \otimes X_2X_2^\top + I)^{-1}(X_1Y_1^\top \otimes X_2Y_2^\top)(Y_1Y_1^\top \otimes Y_2Y_2^\top + I)^{-1}(Y_1X_1^\top \otimes Y_2X_2^\top)C_x \\ &= (I - \frac{1}{2}X_1X_1^\top \otimes X_2X_2^\top)(X_1Y_1^\top \otimes X_2Y_2^\top)(I - \frac{1}{2}Y_1Y_1^\top \otimes Y_2Y_2^\top)(Y_1X_1^\top \otimes Y_2X_2^\top)C_x \\ &= \frac{1}{4}(X_1X_1^\top \otimes X_2X_2^\top)C_x\end{aligned}$$

where we use Sherman Morrison formula in equality, i.e., $(I + uv^\top)^{-1} = I - \frac{1}{2}uv^\top$. It is clear that $C_x = X_1 \otimes X_2$ is the solution. Even when we only have finite sample, C_x will converges to $X_1 \otimes X_2$ as simple size increasing, which illustrates rank-1 approximation is good initial point.

4.5 General TCCA

In this section, we present TCCA in more general setting. Recall the original TCCA problem

$$\min_{\mathcal{U},\mathcal{V}} \frac{1}{n}\sum_{t=1}^{n}(\langle \mathcal{U}, \mathcal{X}_t\rangle - \langle \mathcal{V}, \mathcal{Y}_t\rangle)^2 \text{ such that } \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{U}, \mathcal{X}_t\rangle^2 = 1 = \frac{1}{n}\sum_{t=1}^{n}\langle \mathcal{V}, \mathcal{Y}_t\rangle^2$$

where $\mathcal{U} = U_1 \otimes \cdots \otimes U_m$ and $\mathcal{V} = V_1 \otimes \cdots \otimes V_m$ is rank-one tensor, so TCCA is a way to solve CCA with low rank constraint, which is related to Burer-Monteiro approach in matrix factorization [39, 44, 63, 10, 40, 32, 61, 22]. However, It is well-known that the definition of "low-rank approximation" in tensor setting is not unique. First we can consider add more rank-one tensors, that is,

$$\mathcal{U} = \sum_k a_k U_1^{(k)} \otimes \cdots \otimes U_m^{(k)}, \mathcal{V} = \sum_k b_k V_1^{(k)} \otimes \cdots \otimes V_m^{(k)}$$

which is exactly CP decomposition but such decomposition is ill-posed [15, 11]. Another extension is rank-$(r_1, r_2, \ldots, r_m)$ approximation [13] which is more natural in tensor setting and implies the same result as 2DCCA [30] when $m = 2$.

Form the residual form of TCCA, we know we only need to transform the tensor data $\mathcal{X}_t, \mathcal{Y}_t$ to the same order tensor. If $\mathcal{X}_t \in \mathbb{R}^{d_{x,1}\times\cdots\times d_{x,m}}, \mathcal{Y}_t \in \mathbb{R}^{d_{y,1}\times\cdots\times d_{y,m}}$, let

$$\hat{\mathcal{X}}_t = \mathcal{X}_t \times_1 \mathbf{U}_1 \cdots \times_m \mathbf{U}_m \in \mathbb{R}^{d_1\times\cdots\times d_m}$$

and

$$\hat{\mathcal{Y}}_t = \mathcal{Y}_t \times_1 \mathbf{V}_1 \cdots \times_m \mathbf{V}_m \in \mathbb{R}^{d_1\times\cdots\times d_m}$$

We can rewrite TCCA problem to

$$\min_{\mathbf{U}_i,\mathbf{V}_i \text{ for } i=1,\ldots,m} \frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathcal{X}}_t - \hat{\mathcal{Y}}_t\|_F^2$$
$$\text{such that } \frac{1}{n}\sum_{t=1}^{n}(\mathbf{U}_1\otimes\cdots\otimes\mathbf{U}_m)^\top \text{vec}(\mathcal{X}_t)\text{vec}(\mathcal{X}_t)^\top(\mathbf{U}_1\otimes\cdots\otimes\mathbf{U}_m)=\mathbf{I}$$
$$\frac{1}{n}\sum_{t=1}^{n}(\mathbf{V}_1\otimes\cdots\otimes\mathbf{V}_m)^\top \text{vec}(\mathcal{Y}_t)\text{vec}(\mathcal{Y}_t)^\top(\mathbf{V}_1\otimes\cdots\otimes\mathbf{V}_m)=\mathbf{I}$$

or equivalently

$$\min_{\mathcal{U},\mathcal{V}} \frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathcal{X}}_t - \hat{\mathcal{Y}}_t\|_F^2 \text{ such that for all } j=1,\ldots,m$$
$$\frac{1}{n}\sum_{t=1}^{n}\mathbf{U}_j(\mathcal{X}_t)_{(j)}(\mathbf{U}_1\otimes\ldots\mathbf{U}_{j+1}\otimes\mathbf{U}_{j+1}\cdots\otimes\mathbf{U}_m)^\top(\mathbf{U}_1\otimes\ldots\mathbf{U}_{j+1}\otimes\mathbf{U}_{j+1}\cdots\otimes\mathbf{U}_m)(\mathcal{X}_t)_{(j)}^\top\mathbf{U}_j^\top=\mathbf{I}$$
$$\frac{1}{n}\sum_{t=1}^{n}\mathbf{V}_j(\mathcal{Y}_t)_{(j)}(\mathbf{V}_1\otimes\ldots\mathbf{V}_{j+1}\otimes\mathbf{V}_{j+1}\cdots\otimes\mathbf{V}_m)^\top(\mathbf{V}_1\otimes\ldots\mathbf{V}_{j+1}\otimes\mathbf{V}_{j+1}\cdots\otimes\mathbf{V}_m)(\mathcal{Y}_t)_{(j)}^\top\mathbf{V}_j^\top=\mathbf{I}$$

where $(\mathcal{X}_t)_{(j)}$ is mode-j matricization of $\mathcal{X}_t$ which arranges the mode-j fibers to the columns of the resulting matrix and we utilize the identity for any $j=1,\ldots,m$

$$(\hat{\mathcal{X}}_t)_{(k)} = \mathbf{U}_k(\mathcal{X}_t)_{(k)}(\mathbf{U}_1\otimes\ldots\mathbf{U}_{k+1}\otimes\mathbf{U}_{k+1}\cdots\otimes\mathbf{U}_m)^\top$$

Using the technique of Lagrange multipliers and defining similar notation

$$X_j = (\mathcal{X}_t)_{(j)}(\mathbf{U}_1\otimes\ldots\mathbf{U}_{j+1}\otimes\mathbf{U}_{j+1}\cdots\otimes\mathbf{U}_m)^\top$$

we can derive the high-order power method:

Algorithm 6: Higher-order Power Method

Input : $\{\mathcal{X}_t\}_{t=1}^n$, $\{\mathcal{Y}_t\}_{t=1}^n$
while *not converged* **do**
  for $j=1,2,\ldots,m$ **do**
    Solve the linear system $\mathbf{X}_{kj}^\top\mathbf{X}_{kj}\tilde{\mathbf{U}}_{kj}=\mathbf{X}_{kj}^\top\mathbf{Y}_{kj}\mathbf{V}_{k-1,j}$ to get $\tilde{\mathbf{U}}_{kj}$
    $\mathbf{U}_{kj}=\tilde{\mathbf{U}}_j(\tilde{\mathbf{U}}_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})\tilde{\mathbf{U}}_{kj})^{-1/2}$
    Solve the linear system $\mathbf{Y}_{kj}^\top\mathbf{Y}_{kj}\tilde{\mathbf{U}}_{kj}=\mathbf{X}_{kj}^\top\mathbf{Y}_{kj}\mathbf{U}_{kj}$ to get $\tilde{\mathbf{V}}_{kj}$
    $\mathbf{V}_{kj}=\tilde{\mathbf{V}}_j(\tilde{\mathbf{V}}_{kj}^\top(\frac{1}{n}\mathbf{Y}_{kj}^\top\mathbf{Y}_{kj})\tilde{\mathbf{V}}_{kj})^{-1/2}$
  end
  $k=k+1$
end
Output : $\mathcal{U}_k$, $\mathcal{V}_k$

The only change is just replacing vectors to matrices. Note that we only need to solve SVD for a small matrix, i.g. $\tilde{\mathbf{U}}_{kj}^\top(\frac{1}{n}\mathbf{X}_{kj}^\top\mathbf{X}_{kj})\tilde{\mathbf{U}}_{kj}$.

5 Numerical Studies

5.1 Orthogonal Error

Consider the simplest case for $t = 1, \ldots, n$

$$\begin{aligned} X_t &= c_t(U_1 \otimes U_2) + E'_{xt} \\ Y_t &= c_t(V_1 \otimes V_2) + E'_{yt} \\ E'_{xt} &= E_{xt} - \text{Proj}_{(U_1 \otimes U_2)}(E_{xt}) \\ E'_{yt} &= E_{yt} - \text{Proj}_{(V_1 \otimes V_2)}(E_{yt}) \end{aligned} \tag{18}$$

where $U_1 \in \mathbb{R}^{m_x}, U_2 \in \mathbb{R}^{n_x}, V_1 \in \mathbb{R}^{m_y}, V2 \in \mathbb{R}^{n_y}$ with norm one and $\text{Proj}_V(U)$ is the projection operation projecting U into the subspace generated by V and the elements of E_{xt}, E_{yt} are i.i.d. normal distributed. It is clear that the two solutions of 1DCCA and 2DCCA coincide and the global optimum is 1 and first canonical correlation components of X, Y are $U_1 \otimes U_2$ and $V_1 \otimes V_2$, respectively. Note that we use ALS in our simulations and so components are norm one. Thus, we define the error as

$$error = error(U_1, \hat{U_1}) + error(U_2, \hat{U_2}) + error(V_1, \hat{V_1}) + error(V_2, \hat{V_2}), \text{ where } error(U_1, \hat{U_1}) = 1 - (U_1^\top \hat{U_1})^2$$

and run the simulation in the setting $m_x = m_y = 2, n_x = n_y = 3$ and $n = 15$. The result is following:

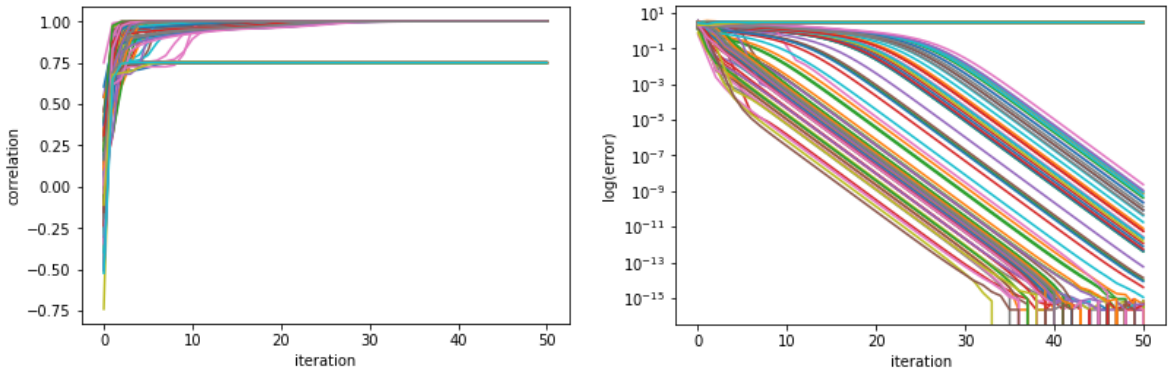


Figure 1: 100 times simulations with random initialization.

Totally 17 simulations are trapped in a local optimum and the linear convergence rate is observed. Next, we try to increase sample size and the result is in Figure 2. Success rate increases and error decreases as sample size increase, which is intuitive. However, surprisingly, even under this simple model that 1DCCA can find optimum, 2DCCA fail due to the non convex nature. Note that even we have 50 samples for only 12 parameters, it is non-negligible chance that 2DCCA find local optimum. SVD-based algorithm have similar behavior with respect to ALS.

5.2 CCA-whitening Models

Consider another models [**?**], for $t = 1, \ldots, n$,

$$\begin{aligned} X_t &= \mathbf{U}_1(\mathbf{\Lambda_1} \odot \mathbf{C}_t + \mathbf{\Lambda}_2 \odot \mathbf{E}_{xt})\mathbf{U}_2^\top \\ Y_t &= \mathbf{V}_1(\mathbf{\Lambda_1} \odot \mathbf{C}_t + \mathbf{\Lambda}_2 \odot \mathbf{E}_{yt})\mathbf{V}_2^\top \end{aligned} \tag{19}$$

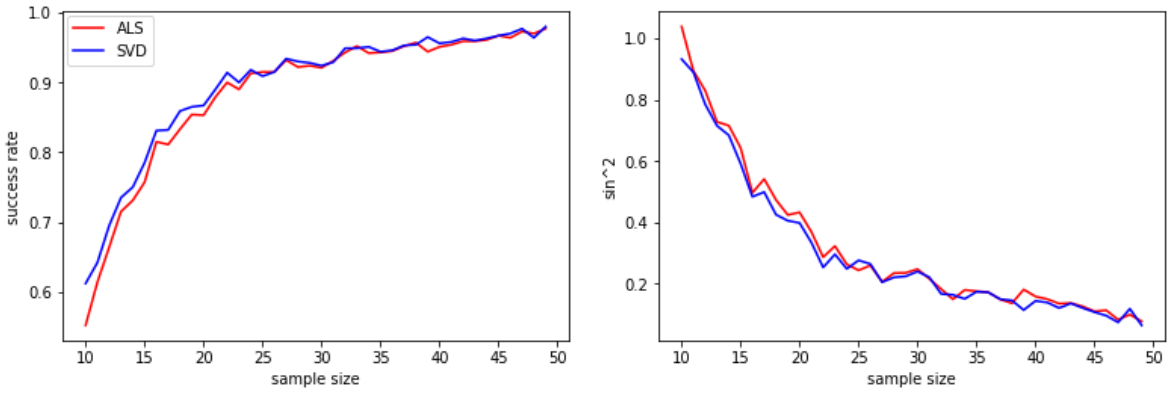


Figure 2: Each points is averaged over 1000 simulation

where $\odot$ is entrywise product (Hadamard product),$\mathbf{U}_1 \in \mathbb{R}^{m_x \times k}, \mathbf{U}_2 \in \mathbb{R}^{n_x \times k}, \mathbf{V}_1 \in \mathbb{R}^{m_y \times k}, \mathbf{V}2 \in \mathbb{R}^{n_y \times k}$, and $\mathbf{\Lambda}$ is a matrix whose element is between 0 and 1. In following simulations, we assume the simple case that $k = 2, m_x = 3, n_x = 4, m_y = 4, n_y = 3$ and the elements of $\mathbf{C}_t, \mathbf{E}_{xt}, \mathbf{E}_{yt}$ are i.i.d. standard normal distributed and

$$\mathbf{\Lambda_1} = \begin{bmatrix} \sqrt{\lambda} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{\Lambda_2} = \begin{bmatrix} \sqrt{1-\lambda} & 1 \\ 1 & 1 \end{bmatrix}$$

It is not hard to see that two population solution of 1DCCA and 2DCCA coincide and the population optimum is λ. The simulation is perform in different sample size $n = 50, 100, 300, 700, 1000, 1500$ and different signal-to-noise $\lambda = 0.8, 0.5, 0.2$, which is conclude in figure 6.

It shows that effective initialization enhance the chance to achieve global optimum and, thus, the average distance between true and sample loadings. Moreover, as the signal-to-noise increasing, the probability of achieving global optimum increases and the optimal correlation calculated by running TCCA with 25 different random initialization approaches to population value.

5.3 Matrix-valued Factor Models

We perform numerical studies to illustrate the benefit of ALS with effective initialization. To impose the tensor low-rank structure to simulation data, we generate data by matrix-valued factor models sharing the same factors [49, 8, 9]. The models are defined as follows:

$$\begin{aligned} \mathbf{X}_t &= \mathbf{L}_x \mathbf{F}_t \mathbf{R}_x + \sigma_x \mathbf{E}_{xt} \\ \mathbf{Y}_t &= \mathbf{L}_y \mathbf{F}_t \mathbf{R}_y + \sigma_y \mathbf{E}_{yt} \quad t = 1, 2, \ldots, n \end{aligned}$$

where F_t is a $k_1 \times k_2$ matrix-valued common factors, $\mathbf{L}_x, \mathbf{L}_y$ are $m_x \times k_1$, $m_y \times k_1$ left loading matrices, $\mathbf{R}_x, \mathbf{R}_y$ are $n_x \times k_2$, $n_y \times k_2$ left loading matrices, and $\mathbf{E}_{xt}, \mathbf{E}_{yt}$ are $m_x \times n_x$, $m_y \times n_y$ error matrices. The entries of $\mathbf{F}_t, \mathbf{E}_{xt}, \mathbf{E}_{yt}$ are sampled from independent normal distribution $N(0, 1)$ and the entries of $\mathbf{L}_x, \mathbf{R}_x, \mathbf{L}_y, \mathbf{R}_y$ are simulated by independent uniform distribution $U(-1, 1)$ and $\sigma_x = 1 = \sigma_y$.

First of all, We demonstrate the convergence property of 2DCCA which estimated by different methods in two setting: $k_1 = 1 = k_2, m_x = 2, n_x = 3, m_y = 3, n_y = 2$ and $k_1 = 2 = k_2, m_x = 2, n_x =$

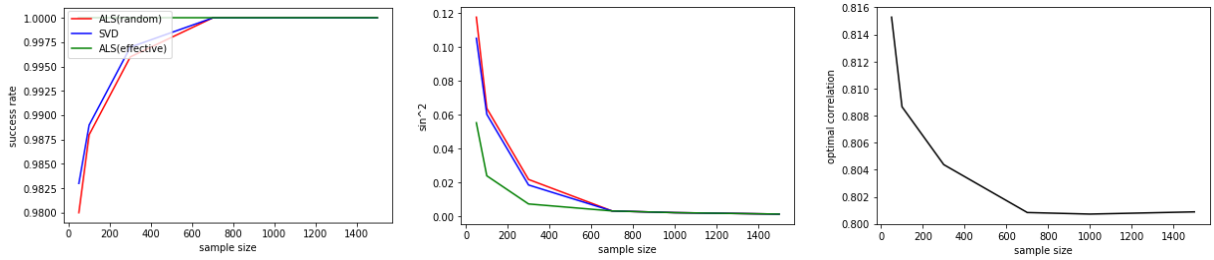


Figure 3: $\lambda = 0.8$

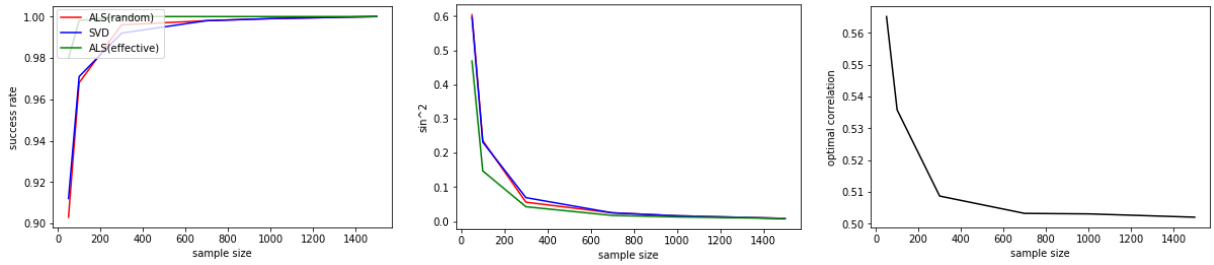


Figure 4: $\lambda = 0.5$

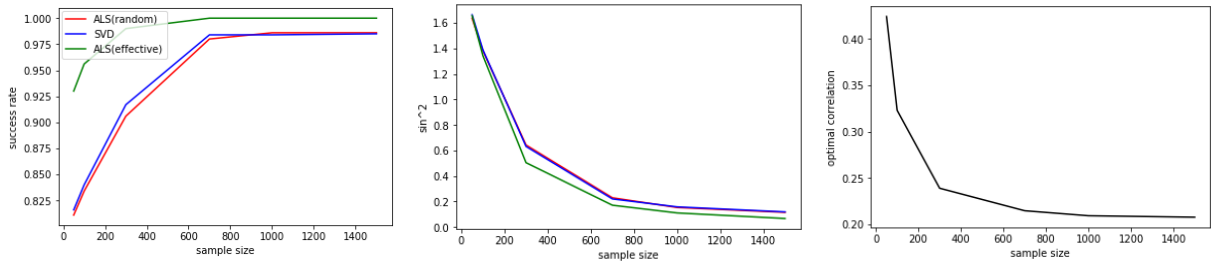


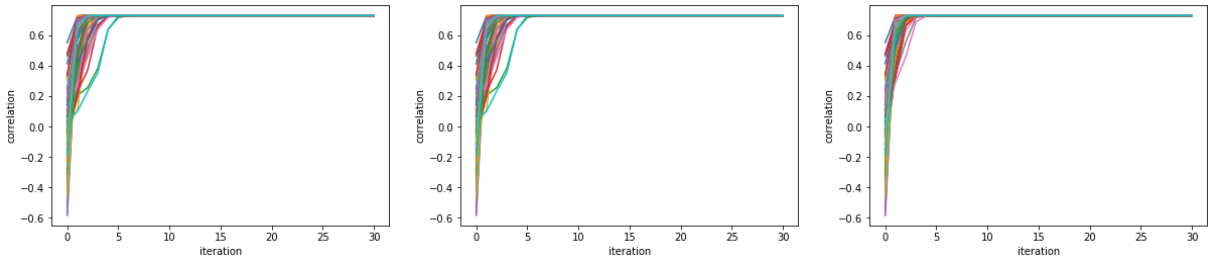
Figure 5: $\lambda = 0.2$

Figure 6: Each points is averaged over 1000 simulation

$3, m_y = 3, n_y = 2$. Each method shares the same random initial points and run 50 times with the same data.

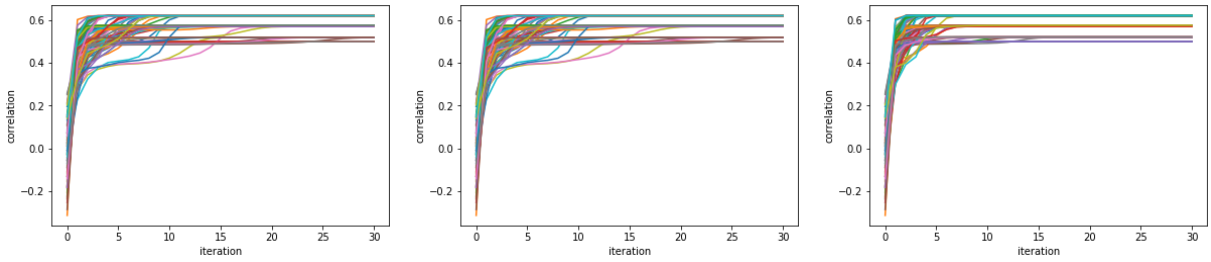
From the figure 7 and 8, ALS and HOPM share the same pattern, which verifies proposition 4.1. The SVD-based method converges slightly faster but it needs to solve eigenproblem in each iteration which has high computation cost when n_x, n_y, m_x, m_y are large.

Next, we test the performance of achieving the global optimum. The case $k_1 = 2 = k_2, m_x = 3, n_x = 5, m_y = 5, n_y = 3$ and $n = 100$ and $n = 300$ for generating data points and three different initialization scheme including fixed initialization, random initialization, and effective initialization are considered. Effective initialization is a scheme of applying the best rank-1 approximation of 1DCCA solution as initialization points. We can see in the figure 7 and 8 that global maximum is found in most initialization. Hence, after data are generated, we run 25 times ALS with random initialization and treat the highest correlation as the global optimum. If the correlations in the different settings are close to the global optimum within some error, say $\epsilon = 0.01$, we count it as a successful outcome to get the global optimum. SVD-based method is also included in our experiment



(a) HOPM (b) ALS (c) SVD-based method

Figure 7: $k_1 = 1 = k_2, m_x = 2, n_x = 3, m_y = 3, n_y = 2$



(a) HOPM (b) ALS (c) SVD-based method

Figure 8: $k_1 = 2 = k_2, m_x = 3, n_x = 5, m_y = 5, n_y = 3$

and results is summarized in 1.

	ALS w/ random initialization	ALS w/ effective initialization	SVD-based algorithm
$n = 100$	10.22%	4.84%	9.60%
$n = 300$	1.16%	0.12%	0.92%

Table 1: Generate 3000 times data and run each method

We can see it is no difference between ALS and SVD-based algorithm and effective initialization improves the probability of achieving global maximum significantly. Also, interestingly but intuitively, as n increasing the probability of achieving increases.

6 Application

In this section, three applications of TCCA are demonstrated for showing the power of reducing computational cost and digging into data.

6.1 Air Pollution in Taiwan

We show how to use TCCA to analyze air pollution data in Taiwan. The question is whether and how geographical and meteorological factors affect air pollution given some quantitative reasoning from TCCA. The monthly average data are downloaded from the website of Environmental Protection Administration Executive Yuan Taiwan. We select the data from 2005 to 2017 for total 156 months,

12 stations, and 7 pollutants. The pollutants include sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), PM10, oxides of nitrogen (NOx), nitric oxide (NO), and nitrogen dioxide (NO2). The data of each pollutant in each station is treated as univariate time series and is removed seasonality by fitted a seasonal ARIMA model. This results in 144 months left and we only eliminate the seasonality effect in the data and analyze the residual.

To examine the geographical factors, we separate Taiwan into north (Guting, Tucheng, Taoyuan and Hsinchu), south (Erlin, Xinying, Xiaogang and Meinong) and east (Yilan, Dongshan, Hualien and Taitung) areas. There are 4 stations in each area (see figure 9). Namely, we have 144 months by 7 pollutants by 4 stations in each area and conduct TCCA for each. Note that to avoid local maximum we repeat TCCA for 20 times and picks the result having highest correlation. The table 2, 3, 4 summarize results.

N vs S	S vs E	N vs E
0.888	0.817	0.904

Table 2: The correlations for each comparison

	N				S			
N vs S	Guting	Tucheng	Taoyuan	Hsinchu	Erlin	Xinying	Xiaogang	Meinong
	0.051	-0.146	-0.032	-0.988	0.777	0.584	-0.125	0.201
	N				E			
N vs E	Guting	Tucheng	Taoyuan	Hsinchu	Yilan	Dongshan	Hualien	Taitung
	0.627	0.671	0.215	0.331	0.895	-0.051	0.441	0.044
	S				E			
S vs E	Erlin	Xinying	Xiaogang	Meinong	Yilan	Dongshan	Hualien	Taitung
	0.625	0.142	0.596	0.483	0.511	-0.145	0.303	0.791

Table 3: The loadings of stations

Table 2 shows that even if east stations are closer to south stations than north overall (figure 9), the correlation between east stations and south stations is smaller than the correlation between north stations and south stations. It may be due to the effect of the Central Mountain Range in Taiwan. Furthermore, we check the loadings of stations and pollutant. Table 3 reveals the location and the role of station and the coefficients show the distance between the stations. In each comparison, the farther a station is, the smaller the magnitude of the coefficient is. For example, Taitung has a higher magnitude in South vs East than North vs East. There is a similar effect in Guting and Yilan. The magnitude of the coefficient of Erlin which surrounded by industrial areas and near thermal power plant is largest in North vs South than South vs East. Another example is Tucheng which also contains an industrial area and many factories, causing a larger coefficient compared to Guting that just next to Tucheng.

The wind condition and raining condition differ in Taiwan. In summer, typhoons and afternoon

	SO2	CO	O3	PM10	NOx	NO	NO2
N	0.001	0.478	-0.322	-0.132	-0.417	0.604	0.333
S	0.571	-0.250	0.415	0.067	0.199	-0.618	0.110
N	-0.779	0.567	0.170	0.165	-0.051	0.112	-0.014
S	-0.027	-0.566	0.402	0.409	0.212	0.010	-0.552
S	-0.270	-0.699	0.146	0.022	0.367	-0.399	-0.351
E	-0.187	-0.292	0.434	0.080	0.219	-0.029	-0.798

Table 4: The loading of pollutants

thunderstorms are common phenomena, which reduces air pollutant concentrations. In contrast, in winter, relatively less rainfall and strongly seasonal wind lead to spreading of pollutants more easily and widely on the whole island. To illustrate this meteorological effect, we separate the sample by summer and winter, i.e., January to March as well as October to December for winter and April to September for summer. The summary is in table 5, 6 and 4.

	N vs S	S vs E	N vs E
Winter	0.940	0.904	0.930
Summer	0.889	0.821	0.914

Table 5: The correlations for each comparisons

Table 5 shows that air pollution in areas are more correlated to each others in winter. PM10 have small coefficients in table 4 and 7, which may indicates that PM10 have different behavior and impact locally.

6.2 Electricity Demands in Adelaide

In this example, we investigate the relationship between the half-hourly electricity demands in Adelaide and temperatures measured at Kent Town from Sunday to Saturday between 7/6/1997 and 3/31/2007, which leads two 508 by 48 by 7 tensors. The value of each half hour at each day is subtracted by the median to remove the seasonality.

First TCCA is performed between the half-hourly electricity demands and temperatures. Table 8, 9 and 10 include the results. we can see the correlation is 0.972989 that is very close to one. The table 10 shows that the magnitude of coefficients in the afternoon (from 2 pm to 4 pm) and evening (from 6 pm to 8 pm) is larger than others and have opposite sign, which inspires us to explore the relationship more carefully. Moreover, the fact revealed in table 9 that the magnitude of coefficients through Wednesday to Saturday are similar is hard to interpret.

In order to explore more sophisticated relation we select the time slot form 10 am to 3 pm and 6 pm to 11 pm respectively. Note that the analysis is not sensitive with respect to the selected time. Table 8 concludes that the correlation between electricity demands and temperatures of daytime is more correlated than the correlation in the nighttime. It is interesting in table 9 that coefficients of loading of the weekday in daytime are similar, and the coefficient of loading of Sunday is largest.

	N				S			
N vs S	Guting	Tucheng	Taoyuan	Hsinchu	Erlin	Xinying	Xiaogang	Meinong
Winter	0.423	0.060	0.078	0.901	-0.925	-0.344	0.161	-0.024
Summer	0.400	-0.277	-0.189	-0.853	0.834	0.435	0.206	0.270
	N				E			
N vs E	Guting	Tucheng	Taoyuan	Hsinchu	Yilan	Dongshan	Hualien	Taitung
Winter	0.599	0.723	0.339	0.060	-0.819	-0.273	-0.500	-0.064
Summer	-0.419	-0.225	-0.783	-0.402	-0.843	-0.195	-0.465	-0.188
	S				E			
S vs E	Erlin	Xinying	Xiaogang	Meinong	Yilan	Dongshan	Hualien	Taitung
Winter	0.961	-0.061	0.116	0.244	-0.462	-0.143	-0.751	-0.449
Summer	0.354	0.798	0.300	0.386	0.733	-0.363	0.332	0.469

Table 6: The loading of stations

Furthermore, when we select time slot 6 pm to 11 pm, Friday, Saturday and Sunday have similar negative coefficients, which is reasonable because we can expect people may have more activities in Fridays night and weekend night that is less correlated to temperature.

6.3 Gene Expression Data

It is well known that genotype data mirror the population structure [6]. Using principal components analysis (PCA), single nucleotide polymorphism (SNP) data of individuals are projected into the first two principal components of the population-genotype matrix and the location information can be recovered. However, this technique cannot simply apply other genomic data types. Recently [7] combines PCA and CCA to overcome this difficulty and reveal population structure in gene expression data.

In [7], they first notice the reason why PCA fails to reconstruct geographical information. It is because the data coming from different laboratories are correlated, and thus regressing the gene expression matrix on different laboratories is conducted to correct confounding. Then they perform PCA on the genotype data to extract first few principal components. Followed by PCA, they use CCA on the batch-corrected expression data and principal components of genotype data, which succeeds to separate the population in expression data (Figure 10 (a)). Intuitively, principal components of genotype data are clustered by population, and thus, guide the expression data to split by population via CCA. It is not surprising to see the distinct population patterns in CCA projection of the expression data. PCA not only group the genotype data but also reduce the computational cost. This is essential because the original genotype data contain around 7 million SNPs. To utilize the information as much as possible, our goal here is using the CCA without PCA to the same separation result. The trick is to reshape the expression data and genotype data to the matrix and perform TCCA.

To demonstrate the computation benefit of TCCA, we use 318 individuals with genotype data

	SO2	CO	O3	PM10	NOx	NO	NO2
N(Winter)	0.122	0.672	-0.176	-0.084	0.298	-0.083	-0.633
S(Winter)	0.228	0.805	0.357	0.071	0.292	-0.242	0.152
N(Summer)	-0.540	-0.344	0.532	0.060	0.263	-0.480	0.066
S(Summer)	-0.000	0.136	-0.360	0.003	0.232	0.740	-0.499
N(Winter)	0.477	0.330	-0.247	-0.078	0.397	-0.430	-0.504
E(Winter)	-0.558	-0.228	0.333	0.150	0.417	-0.484	-0.307
N(Summer)	-0.111	0.113	0.074	0.188	0.450	-0.283	-0.807
E(Summer)	0.272	-0.371	0.127	0.209	0.099	0.473	-0.704
S(Winter)	0.404	0.027	-0.263	-0.019	-0.627	0.392	0.469
E(Winter)	-0.304	-0.523	0.171	0.077	0.280	-0.712	-0.115
S(Summer)	-0.608	-0.103	0.174	0.009	0.455	-0.300	-0.541
E(Summer)	-0.111	0.024	0.269	0.046	0.262	0.619	-0.679

Table 7: The loading of pollutants

A(0am-24pm)	D(10am-3pm)	N(6pm-11pm)
0.973	0.885	0.714

Table 8: correlation between electricity demands and temperatures

from 1000 genomes phase 1 and corresponding RNA-seq data from GEUVADIS in 4 population, GBR, FIN, YRI and TSI, which is same as [7] for comparison. We follow the same procedure in [7] to extract expression data and remove the confounding. This left 14079 genes in expression data and reshape to a 361×39 matrix. The Phase 1 1000 genomes genotypes contain 39728178 variants. We use LD pruning which uses a moving window to compute pairwise correlation and removes high correlation SNPs. This left 738192 SNPs and reshape to a 1014×728 matrix. Then we perform TCCA and the result is shown in figure 10 (b). We can see that TCCA improves the separation. Thanks to more information fewer points in the overlapping region and points of each population are more concentrated. Moreover, our method do not require select the number o fPCA components.

7 Conclusion

In this paper, we extend 2DCCA to tensor-valued setting and foster a deeper understanding of 2DCCA and TCCA. In particular, we first provide two algorithms, OHPM and ALS, which are compatible with SVD-based algorithm proposed in the original 2DCCA paper but more suitable for large scale data. Convergence properties are proved and an error bound for inexact updating is provided. All results are also justified by a sequence of simulations in different models and parameters. Finally, three real data sets are utilized to demonstrate the ability of using low rank structure and computational effectiveness, showing superior performance and great potential of TCCA.

	monday	tuesday	wednesday	thursday	friday	saturday	sunday
whole day	0.240	0.413	0.385	0.436	0.420	0.441	0.244
day (10am-3pm)	0.375	0.437	0.440	0.432	0.396	0.299	0.198
night (6am-11pm)	0.195	0.325	0.528	0.160	-0.323	-0.584	-0.322

Table 9: Loadings of days

0		1		2		3	
-0.080	-0.076	-0.104	-0.104	-0.061	-0.021	0.021	0.067
4		5		6		7	
0.099	0.121	0.12	0.075	-0.002	-0.086	-0.129	-0.144
8		9		10		11	
-0.155	-0.156	-0.151	-0.125	-0.064	0.002	0.066	0.116
12		13		14		15	
0.145	0.174	0.179	0.177	0.206	0.238	0.262	0.284
16		17		18		19	
0.281	0.227	0.121	-0.054	-0.231	-0.237	-0.214	-0.206
20		21		22		23	
-0.135	-0.071	-0.058	-0.069	-0.080	-0.055	-0.004	-0.062

Table 10: Loadings of time in all day

References

[1] P. a. Absil, R. Mahony, and B. Andrews. Convergence of the Iterates of Descent Methods for Analytic Cost Functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Doubly Accelerated Methods for Faster CCA and Generalized Eigendecomposition. pages 1–35, 2016.

[3] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep Canonical Correlation Analysis. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 28:1247–1255, 2013.

[4] Raman Arora, Teodor V Marinov, Poorya Mianjy, and Nathan Srebro. Stochastic Approximation for Canonical Correlation Analysis. (1):1–10, 2017.

[5] Francis R Bach and Michael I Jordan. A Probabilistic Interpretation of Canonical Correlation Analysis. *Dept Statist Univ California Berkeley CA Tech Rep*, 688:1–11, 2006.

[6] Adam R Boyko, Adam Auton, John Novembre, Toby Johnson, Katarzyna Bryc, Amit Indap, Karen S King, Sven Bergmann, Matthew R Nelson, Matthew Stephens, and Carlos D Bustamante. Genes mirror geography within Europe. 456(November), 2008.

[7] Brielin C Brown and Nicolas L Bray. Expression reflects population structure. 2018.

[8] Elynn Yi Chen and Rong Chen. Factor Models for High-Dimensional Dynamic Networks: with Application to International Trade Flow Time Series 1981-2015. pages 1–24, 2017.

[9] Yi Chen, Ruey S Tsay, and Rong Chen. Constrained factor models for high-dimensional matrix-variate time series. 2017.

10		11		12
-0.611	-0.385	-0.165	-0.084	0.012
	13		14	
0.198	0.281	0.264	0.321	0.391
18		19		20
0.676	0.523	0.384	0.275	0.167
	21		22	
0.109	0.066	0.033	0.025	0.029

Table 11: loadings of daytime and nighttime

[10] Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. pages 1–63, 2015.

[11] Pierre Comon, Xavier Luciani, Pierre Comon, and Xavier Luciani. Tensor Decompositions , Alternating Least Squares and other Tales To cite this version : Tensor Decompositions , Alternating Least Squares and other Tales. 23:393–405, 2009.

[12] L De Lathauwer, B De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[13] L De Lathauwer, B L R De Moor, and J Vandewalle. On the Best Rank-1 and Rank-(${R1,R2,,RN}$) Approximation of Higher-Order Tensors. 21(4):1324–1342, 2000.

[14] Christopher De Sa, Bryan He, Ioannis Mitliagkas, Christopher Ré, and Peng Xu. Accelerated Stochastic Power Iteration. 2017.

[15] Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. pages 1–44, 2006.

[16] Jr. E. P. Miles. Generalized Fibonacci Numbers and Associated Matrices. *The American Mathematical Monthly*, 67:745–752, 1960.

[17] Xiao Fu, Kejun Huang, Mingyi Hong, Nicholas D. Sidiropoulos, and Anthony Man Cho So. Scalable and Flexible Multiview MAX-VAR Canonical Correlation Analysis. *IEEE Transactions on Signal Processing*, 65(16):4150–4165, 2017.

[18] Kenji Fukumizu, Francis R Bach, and A Gretton. Statistical Consistency of Kernel Canonical Correlation Analysis. *Journal of Machine Learning Research*, 8:361–383, 2007.

[19] Chao Gao, Dan Garber, Nathan Srebro, Jialei Wang, and Weiran Wang. Stochastic Canonical Correlation Analysis. 1(4):1–35, 2017.

[20] Rong Ge, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Efficient Algorithms for Large-scale Generalized Eigenvector Computation and Canonical Correlation Analysis. pages 1–26, 2016.

[21] Yu Guan, Moody T. Chu, and Delin Chu. Convergence analysis of an SVD-based algorithm for the best rank-1 tensor approximation. *Linear Algebra and Its Applications*, 555:53–69, 2018.

[22] Benjamin D Haeffele and Rene Vidal. Structured Low-Rank Matrix Factorization: Global Optimality, Algorithms, and Applications. 14(8):1–19, 2017.

[23] David R Hardoon and John Shawe-Taylor. Convergence analysis of kernel Canonical Correlation Analysis: theory and practice. *Mach Learn*, 74:23–38, 2009.

[24] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664, 2004.

[25] Harold Hotelling. Relations Between Two Sets of Variates. 28(3):321–377, 1936.

[26] Cheng Jin, Wenhui Mao, Ruiqi Zhang, Yuejie Zhang, and Xiangyang Xue. Cross-Modal Image Clustering via Canonical Correlation Analysis. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 151–159, 2015.

[27] Dan Kalman. Generalized Fibonacci Numbers By Matrix Methods. *The Fibonacci Quarterly*, (2):3–6, 1982.

[28] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor Canonical Correlation Analysis for Action Classification. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[29] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[30] Sun Ho Lee and Seungjin Choi. Two-dimensional canonical correlation analysis. *IEEE Signal Processing Letters*, 2007.

[31] Xingguo Li, Zhehui Chen, Lin Yang, Jarvis Haupt, and Tuo Zhao. Online Generalized Eigenvalue Decomposition: Primal Dual Geometry and Inverse-Free Stochastic Optimization. (2), 2017.

[32] Yuanxin Li, Cong Ma, Yuxin Chen, and Yuejie Chi. Nonconvex Matrix Factorization from Rank-One Measurements. pages 1–33, 2018.

[33] Zhening Li, André Uschmajew, and Shuzhong Zhang. On convergence of the maximum block improvment method. *SIAM Journal on optimization*, 25(1):210–233, 2015.

[34] David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schölkopf. Randomized Nonlinear Component Analysis. 32, 2014.

[35] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. Tensor canonical correlation analysis for multi-view dimension reduction. In *2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016*, pages 1460–1461, 2016.

[36] Zhuang Ma, Yichao Lu, and Dean Foster. Finding Linear Structure in Large Datasets with Scalable Canonical Correlation Analysis. 2015.

[37] Tomer Michaeli, Weiran Wang, and Karen Livescu. Nonparametric Canonical Correlation Analysis. *International Conference on Machine Learning*, pages 1967—-1976, 2015.

[38] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Asymmetrically Weighted CCA And Hierarchical Kernel Sentence Embedding For Image & Text Retrieval. *arXiv*, 2016.

[39] Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding Low-Rank Solutions via Non-Convex Matrix Factorization, Efficiently and Provably. (1):1–45, 2016.

[40] Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. 2016.

[41] Mehran Safayani, Seyed Hashem Ahmadi, Homayun Afrabandpey, and Abdolreza Mirzaei. An EM based probabilistic two-dimensional CCA with application to face recognition. *Applied Intelligence*, 48(3):755–770, 2018.

[42] Mehran Safayani and Saeid Momenzadeh. A Latent Variable Model for Two-Dimensional Canonical Correlation Analysis and its Variational Inference. 2017.

[43] Reinhold Schneider and André Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via Lojasiewicz inequality. *SIAM Journal on Optimization*, 25(1):622–646, 2015.

[44] Chuangchuang Sun and Ran Dai. A decomposition method for nonconvex quadratically constrained quadratic programs. In *Proceedings of the American Control Conference*, pages 4631–4636, 2017.

[45] Tingkai Sun and Songcan Chen. Class label versus sample label-based CCA. *Applied Mathematics and Computation*, 185(1):272–283, 2007.

[46] Xiaotong Suo, Victor Minden, Bradley Nelson, Robert Tibshirani, and Michael Saunders. Sparse canonical correlation analysis. pages 1–20, 2017.

[47] Kean Ming Tan, Zhaoran Wang, Han Liu, and Tong Zhang. Sparse Generalized Eigenvalue Problem: Optimal Statistical Rates via Truncated Rayleigh Flow. pages 1–29, 2016.

[48] André Uschmajew. A new convergence proof for the high-order power method and generalizations. *Preprint*, 33(2):1–10, 2014.

[49] Dong Wang, Xialu Liu, and Rong Chen. Factor Models for Matrix-Valued High-Dimensional Time Series. 2016.

[50] Jialei Wang, Weiran Wang, Dan Garber, and Nathan Srebro. Efficient coordinate-wise leading eigenvector computation. 1, 2017.

[51] Weiran Wang, Jialei Wang, Dan Garber, and Nathan Srebro. Efficient Globally Convergent Stochastic Optimization for Canonical Correlation Analysis. (Nips):1–25, 2016.

[52] Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep Variational Canonical Correlation Analysis. 1, 2016.

[53] Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[54] D A Wolfram. Solving generalized Fibonacci recurrences. (June 1996):129–145, 1998.

[55] Chang Xu, Dacheng Tao, and Chao Xu. A Survey on Multi-view Learning. pages 1–59, 2013.

[56] Yangyang Xu and Wotao Yin. A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[57] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-Dimensional {PCA}:A New Approach to Appearance-Based Face Representation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[58] Mao-Long Yang, Quan-Sen Sun, and De-Shen Xia. Two-Dimensional Partial Least Squares and Its Application In Image Recognition. *Intelligent Computing Techniques*, pages 208–215, 2008.

[59] Jieping Ye, Ravi Janardan, and Qi Li. Two-Dimensional Linear Discriminant Analysis. In L K Saul, Y Weiss, and L Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1569–1576. MIT Press, 2005.

[60] Florian Yger, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Adaptive canonical correlation analysis based On matrix manifolds. *29th International Conference on Machine Learning*, (2006):1071–1078, 2012.

[61] Ming Yu, Zhaoran Wang, Varun Gupta, and Mladen Kolar. Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. 2018.

[62] Daoqiang Zhang and Zhi Hua Zhou. (2D)2 PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1-3):224–231, 2005.

[63] Tuo Zhao, Zhaoran Wang, and Han Liu. A Nonconvex Optimization Framework for Low Rank Matrix Estimation. In *Advances in Neural Information Processing Systems*, pages 1–9, 2015.

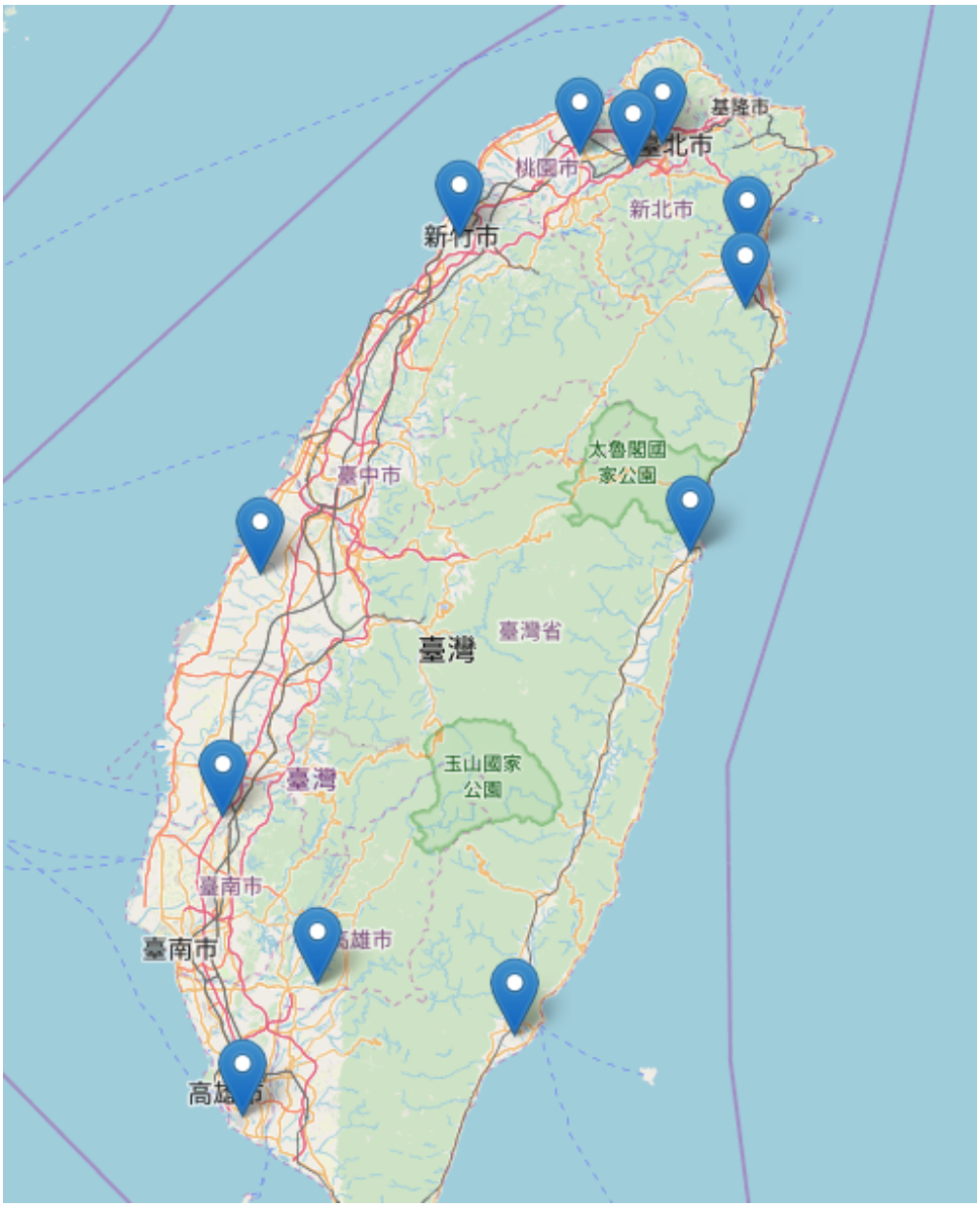
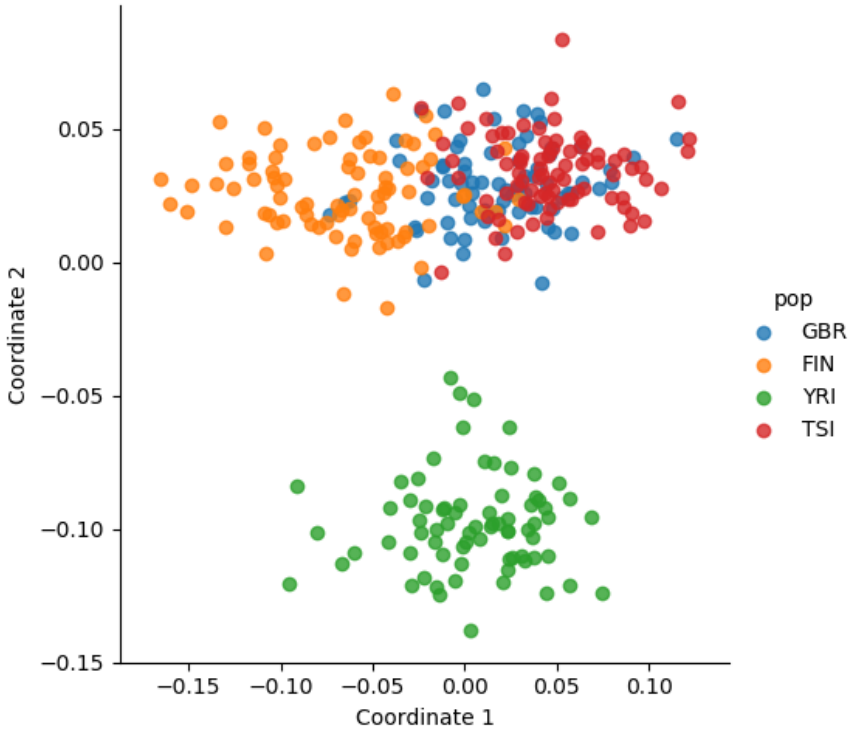
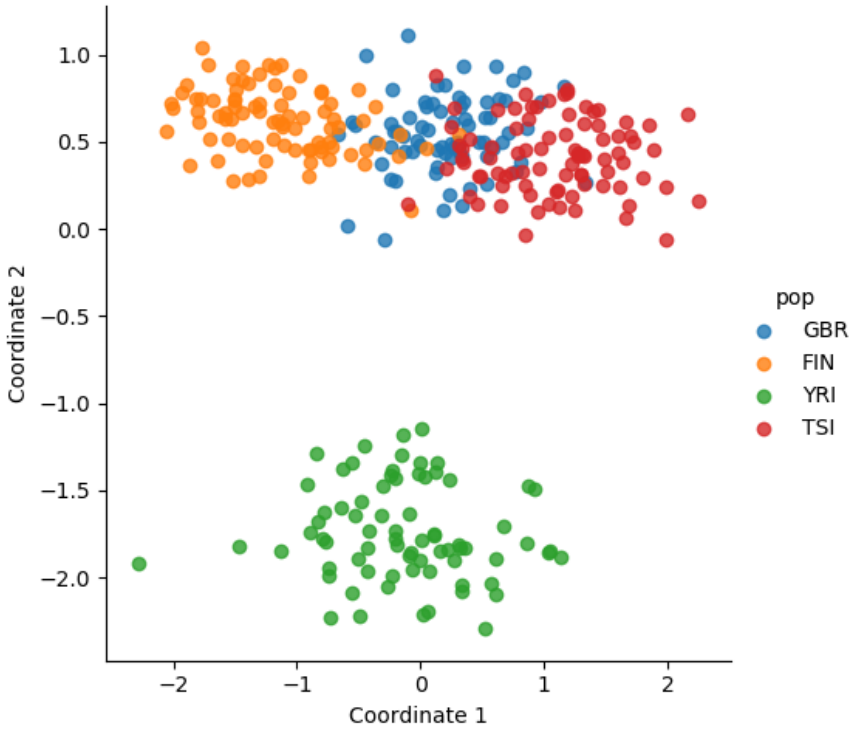


Figure 9: Stations in Taiwan



(a) PCA+CCA



(b) TCCA

Figure 10: The population structure of geno expression data