Neurocomputing 190 (2016) 25–34

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient dimension reduction for high-dimensional matrix-valued data

Dong Wang [a,b,*], Haipeng Shen [c], Young Truong [d]

[a] Department of Statistics and Biostatistics, Rutgers University, Piscataway, NJ 08854, USA
[b] Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA
[c] Innovation and Information Management, School of Business, Faculty of Business and Economics, The University of Hong Kong, Hong Kong, China
[d] Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

ARTICLE INFO

Article history:
Received 24 December 2014
Received in revised form
10 October 2015
Accepted 30 December 2015
Communicated by Qi Li
Available online 9 January 2016

Keywords:
Dimension reduction
Matrix-valued data
Singular value decomposition

ABSTRACT

Collection of groups of high-dimensional matrix-valued data is becoming increasingly common in many modern applications such as imaging analysis. The massive size of such data creates challenges in terms of computing speed and computer memory. Numerical techniques developed for small or moderate-sized datasets simply do not translate to such massive datasets. The need to analyze such data effectively calls for the development of efficient dimension reduction techniques. We propose a novel dimension reduction approach that has nice approximation property, computes fast for high dimensionality, and also explicitly incorporates the intrinsic two-dimensional structure of the matrices. We approximate each matrix as the product of a group-level left basis matrix, a group-level right basis matrix, and an individual-level coefficient matrix, which are estimated through a two-stage singular value decomposition. We discuss the connection of our proposal with existing approaches, and compare them both numerically and theoretically. We also obtain theoretical upper bounds on the approximation error of our method. In the numerical studies, ours is much faster than the most accurate one, comparable to the near-optimal one both computationally and theoretically, and more precise than the one that requires the same amount of memory.

1. Introduction

As the technology advances, matrix-valued data are more and more common. For instance, a typical *functional magnetic resonance imaging* (fMRI) dataset is usually represented as a group of matrices of the same size, where each matrix is the measurement of the blood oxygen level dependent contrast for one subject, with each column corresponding to a vectorized three-dimensional image at a certain time point, and each row being a sequence of temporal observations for a particular brain voxel.

These matrices are often of high or even ultra-high dimension that needs a large amount of memory. For instance, a collection of fMRI data for 100 subjects may consist of 100 matrices with the spatial dimension corresponding to as many as 200,000 voxels and the temporal dimension consisting of around 200 time points, which altogether requires about 30 gigabytes (GB) memory in double precision. Hence, it is crucial to develop a group-wise dimension reduction technique that is precise and scales well for high-dimensional data, which is the goal of the current paper.

Most conventional dimension reduction techniques were developed for groups of vector-valued data, such as the popular principal component analysis (PCA) [1]. To apply these approaches directly to matrix-valued data, we need to vectorize each matrix. The conventional one dimensional (1D) PCA then projects vector-valued observations onto a set of orthogonal directions that preserve the maximum amount of variation in the data. These directions are characterized by the leading eigenvectors of the sample covariance matrix. However, the vectorization ignores the intrinsic two-dimensional (2D) structure embedded in the matrices, and creates high-dimensional vectors that increase computational/memory burden. This usually makes the follow-up dimension reduction not efficient.

Several dimension reduction methods have been developed that incorporate the 2D structure of matrices. Motivated by 1DPCA, 2DPCA of Yang et al. [2] projects each matrix onto the principal eigen-space of the row–row covariance matrix without vectorization. 2DPCA can also be understood through the perspective of a one-sided-type low rank approximation to matrices. However, 2DPCA only takes into consideration the row–row covariance matrix. To fully capture both the row–row and column–column correlations, Ye [3] proposed the *generalized low rank approximations of matrices* (GLRAM) approach which is a two-sided-type low rank approximation. The idea of GLRAM originates from the

[*] Corresponding author at: Department of Statistics and Biostatistics, Hill Center, Rutgers University, Piscataway, NJ 08854, USA.
E-mail addresses: dong.v.wang@rutgers.edu (D. Wang), haipeng@hku.hk (H. Shen), truong@bios.unc.edu (Y. Truong).

http://dx.doi.org/10.1016/j.neucom.2015.12.096
0925-2312/© 2016 Elsevier B.V. All rights reserved.

minimization of the sum of squared residuals. The optimization criterion has no closed form solution and naturally leads to an iterative algorithm that can be slow. To achieve better computational efficiency, Ding and Ye [4] proposed a non-iterative algorithm named *two-dimensional singular value decomposition* (2DSVD) which only implements eigen-decomposition on the row–row and column–column covariance matrices. Zhang and Zhou [5] independently proposed *two-directional two-dimensional principal component analysis* ($(2D)^2$PCA) that is intrinsically equivalent to 2DSVD. The reduction of the computation cost inevitably makes the reconstruction error of 2DSVD and $(2D)^2$PCA larger than GLRAM, the optimal iterative procedure.

Besides computational speed, the limitation of the computer memory is another major hurdle that one has to tackle when analyzing massive data. Take the aforementioned fMRI data for example. The large amount of memory needed is beyond general computer capacity and the various algorithms discussed above are hence not implementable.

To cope with memory-demanding data and further speed up the computation, recently Crainiceanu et al. [6] proposed the *population value decomposition* (PVD) approach that essentially boils down to a two-step *singular value decomposition* (SVD) algorithm. In the first step, SVDs are applied separately to individual matrices that are of relatively small size, and the leading left and right singular vectors are retained. This can be performed either in parallel or sequentially and often requires much less memory. In the second step, the leading left and right singular vectors obtained in the first step are concatenated column-wise, respectively; and SVD is applied again to each concatenated matrix. These aggregated matrices are substantially smaller than the raw data matrices if one only keeps the few leading singular vectors. The resulting left singular vectors in the second step are used to obtain the final approximation for the original matrices. Obviously, ignoring the higher-order singular vectors in the first step results in less accuracy for PVD. But PVD effectively reduces the computational burden, and is applicable for high-dimensional matrices. Recently Eloyan et al. [7] incorporated PVD nicely into a likelihood-based independent component analysis framework.

One drawback of PVD though is that the computational efficiency does come at the price of reduced approximation accuracy. In this paper, we further improve PVD and develop an *adjusted PVD* (APVD) algorithm that has the same computational cost and requires the same amount of memory as PVD, but produces more precise results. In fact, APVD often performs as accurate as GLRAM and 2DSVD for matrices of small to moderate sizes when they can be computed.

The key idea of the APVD modification arises from the observation that PVD assigns equal weights in the group-level SVD to those leading singular vectors obtained in the first SVD step. We all know that the singular vectors have a natural order of relative importance as reflected by the corresponding singular values. Hence, we adjust PVD by incorporating the relative importance of the singular vectors, which indeed results in a more accurate estimation of the group components. The first step of APVD is the same as PVD. While in the second step, our APVD procedure concatenates the scaled singular vectors, i.e. the product of the singular vectors and their corresponding singular values from each individual matrix, instead of concatenating just the singular vectors. Furthermore, we establish theoretical justification for APVD in terms of upper bound on the normalized reconstruction errors.

The rest of this paper is organized as follows. In Section 2, we state the model and give a brief review of the GLRAM, 2DSVD, and PVD procedures. In Section 3, we then describe the APVD algorithm, and compare the computational complexities of the various approaches along with their connections. The theoretical properties of APVD are studied in Section 4. Numerical comparisons through simulation studies and a classical face image dataset are presented in Section 5, to show that APVD performs comparable to GLRAM and 2DSVD and better than PVD. We conclude in Section 6, and relegate all proofs to Appendix A.

2. Preliminaries

2.1. The model

Consider there are I matrices of dimension $m \times n$, denoted as X_i, $i=1,\ldots,I$. To achieve group dimension reduction for the matrices, a reasonable model can be written as

$$X_i = LW_iR^T + E_i, \tag{1}$$

where $L \in \mathbb{R}^{m\times r_L}$ and $R \in \mathbb{R}^{n\times r_R}$ are orthonormal matrices representing the left and right group components, respectively, $W_i \in \mathbb{R}^{r_L\times r_R}$ is the individual coefficient matrix, and the error matrix E_i contains the individual approximation errors. Throughout this paper, we assume that $\sum_{i=1}^{I} X_i = 0$. If we choose $r_L \ll n$ and $r_R \ll m$, the size of the individual component W_i (r_Lr_R) is much smaller than the size of the original data (mn), which achieves the goal of dimension reduction.

The decomposition of X_i as in (1) is closely related to the SVD of a single matrix. Suppose $I=1$, that is, there is only one matrix. Then the optimal L and R that minimize the sum of squares of the approximation errors in E_i are the $r=\min(r_L, r_R)$ leading left and right singular vectors of X_1, and W_1 can always be required to be a diagonal matrix with the r leading singular values. When $I>1$, Model (1) relaxes the requirement that all of the subject-specific terms W_i should be diagonal matrices and only keep the orthonormal constraints of the group components. The reason is that the subspace spanned by the columns of L (or R) can be thought of as the best rank r_L (or r_R) subspace that spans the column (or row) subspace of all the X_i's; the W_i's are the coefficients when projecting X_i onto L and R, which are not necessarily diagonal matrices.

2.2. Review of existing methods

The GLRAM, 2DSVD, PVD (and APVD) procedures offer different ways of estimating Model (1), as we shall review below. Least squares offers a natural criterion for model estimation. It can be shown that the least squares estimator of W_i is given by $\widehat{W_i} = \hat{L}^T X_i \hat{R}$, once we obtain the group component estimates, $\hat{L}$ and $\hat{R}$. Therefore, for the rest of this paper, we focus on the estimation of L and R. Moreover, for simplicity, we describe how each approach can be used to estimate the left component L; R can be estimated in the same way using the transpose of X_i.

The GLRAM of Ye [3] borrows the minimum reconstruction error property of SVD and seeks L, R and W_i to minimize the reconstruction error in the least squares sense:

$$\begin{aligned} &\min_{L,R,W_i} \sum_{i=1}^{I} \|X_i - LW_iR^T\|_F^2, \\ &\quad \text{s.t. } R^TR = I_{r_R}, L^TL = I_{r_L}, \\ &\quad L \in \mathbb{R}^{m\times r_L},\ R \in \mathbb{R}^{n\times r_R} \text{ and } W_i \in \mathbb{R}^{r_L\times r_R}, \end{aligned} \tag{2}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm.

Ye [3] pointed out that the optimization problem (2) has no closed form solutions for L and R. Hence, GLRAM solves the problem in an iterative fashion. In each iteration, it alternates the updating of L (or R) as the leading r_L (or r_R) eigenvectors of $\sum_{i=1}^{I} X_iRR^TX_i^T$ (or $\sum_{i=1}^{I} X_i^TLL^TX_i$), by fixing R (or L) as the corresponding estimate obtained in the previous iteration. The algorithm terminates until it reaches a certain convergence criterion.