Paper Project

Sparse Higher-Order Principal Components Analysis *

Woo Min Kim

May. 2. 2018

1 Introduction

The Principal Component Analysis (PCA) has been widely utilized to reduce dimension or to select features by finding major variation of the data. And as an extension of PCA on matrices, a more general approach such as PCA on high-dimensional tensor or multi-modal data has become more attractive because of the appearance of multi-scale structured data such as human connectome data. Studies on higher-order PCA were done by Kolda and Bader (2009) by exploiting various methods such as the CANDECOMP/PARAFAC (CP) by Harshman (1970) and the Tucker decomposition by Tucker (1966). Many studies on theses decompositions were done; however, relatively few focused on the sparsity in the factors. Sparsity here means the approximation with q principal components where $q < p$ and p is number of factors.

Sparsity in tensor decompositions is appealing with several reasons. First, with tensor decompositions, a large high-dimensional data can be compressed tremendously. Second, in high-dimensional settings, many features are often uncorrelated, and sparsity gives one an automatic tool for feature selection in high-dimensional tensors. Third, although PCA on matrix data is asymptotically inconsistent with high dimensionality, there has been studies shows that sparsity in PCA leads to consistent principal component directions. Fourth, high-dimensional tensors are not proper for a nice visualization; however, sparsity in PCA limits the number of features and leads to better visualization and interpretation on exploratory data analysis.

The author introduced two original algorithms to incorporate sparsity into higher-order PCA: the $Sparse\ Higher-Order\ SVD$ and the $Sparse\ CP\ Decomposition$.

The notation was maintained as in Kolda and Bader (2009). $\boldsymbol{\mathcal{X}}$, $\mathbf{X}$, $\mathbf{x}$, and x are tensors, matrices, vectors and scalars, respectively. The outer product and the Tucker product will be denoted by $\circ$ and $\times_i$; the subscript i refers to the mode being multiplied using regular matrix multiplication. $X_{(i)}$ denotes the matricization of tensor; in other words, if $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q}$, then $X_{(1)} \in \mathbb{R}^{n \times pq}$. $||\boldsymbol{\mathcal{X}}_F||$ is the tensor Frobenius norm and it is defined as follows: $||\boldsymbol{\mathcal{X}}_F|| = \sqrt{\sum_i \sum_j \sum_k \boldsymbol{\mathcal{X}}_{ijk}^2}$. Only three-mode tensor was used for the notational simplicity.

*Sparse Higher-Order Principal Components Analysis (G. Allen 2012)

2 Methods

2.1 Sparse Higher-Order SVD

Higher-order principal components can be easily obtained by the higher-order SVD (HOSVD or Tucker decomposition). For the simplicity, the author used three-mode tensor, and the dimensions can also be freely changed. With a three-mode tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q}$, the decomposition can be expressed as $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{D}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$ where the factors $\mathbf{U} \in \mathbb{R}^{n \times K_1}$, $\mathbf{V} \in \mathbb{R}^{p \times K_2}$, $\mathbf{W} \in \mathbb{R}^{q \times K_3}$ are orthonormal and $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{K_1 \times K_2 \times K_3}$ is the core tensor.

Then, with the following algorithm, the Sparse Higher-Order SVD can be done:

1. Compute $\mathbf{U}$: the first K_1 principal components of $\mathbf{X}_{(1)} \in \mathbb{R}^{n \times pq}$
2. Compute $\mathbf{V}$: the first K_2 principal components of $\mathbf{X}_{(2)} \in \mathbb{R}^{p \times nq}$
3. Compute $\mathbf{W}$: the first K_3 principal components of $\mathbf{X}_{(3)} \in \mathbb{R}^{q \times np}$
4. Compute $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{D}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$

With a three-mode tensor, the HOSVD can be obtained by performing PCA three times on data matricized along each of the three dimensions. Then sparse HOSVD can be achieved by replacing PCA with sparse PCA to obtain sparse factors for each tensor mode. However, this method is not appealing and has drawbacks. First, any optimization problem with a loss function is not considered. Second, in high-dimensional settings, this method is computationally intensive. Thus, even though the sparse HOSVD is quite intuitive and simple, it is not appealing. Instead of this method, the author introduced sparse CP decomposition algorithm in the following section.

2.2 Sparse CANDECOMP/PARAFAC Decomposition

As explained in the Appendix section, the CP decomposition finds a tensor in the form of a sum of rank on tensors: $\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} d_k \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$ where $\mathbf{u}_k \in \mathbb{R}^n$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{w}_k \in \mathbb{R}^q$ and $d_k \geq 0$. Prior to explaining sparse HOPCA algorithm by Sparse CP decomposition, the Tensor Power Method should be introduced first which is used for the single-factor CP problem.

2.3 Tensor Power Method

The single-factor CP decomposition leads one to solve the following optimization problem:

$$\text{maximize}_{\mathbf{u},\mathbf{v},\mathbf{w}} \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}$$

$$\text{subject to } \mathbf{u}^T\mathbf{u} = 1, \mathbf{v}^T\mathbf{v} = 1, \mathbf{w}^T\mathbf{w} = 1. \tag{1}$$

Then the Lagrangian is given by $L(\mathbf{u}, \gamma) = (\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}) \times_1 \mathbf{u} - \gamma(\mathbf{u}^T\mathbf{u} - 1)$. Then we have $\hat{\mathbf{u}} = \frac{\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}}{2\hat{\gamma}}$ and $\hat{\gamma} = ||\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}||_F$ because $\hat{\mathbf{u}}^T\hat{\mathbf{u}} = 1$. Likewise we can obtain $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$. Then we have:

$$\hat{\mathbf{u}} = \frac{\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v} \times_3 \mathbf{w}}{||\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v} \times_3 \mathbf{w}||_2}, \ \hat{\mathbf{v}} = \frac{\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u} \times_3 \mathbf{w}}{||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u} \times_3 \mathbf{w}||_2} \text{ and } \hat{\mathbf{w}} = \frac{\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u} \times_2 \mathbf{v}}{||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u} \times_2 \mathbf{v}||_2}.$$

By updating each coordinate, the object function increases and since the it is bounded above by d, the convergence is assured. It is true that this approach only converges to a local optimum; however, all other algorithmic approaches for CP problem converges to a local optimum as well. Now, we have the solution for single-factor CP decomposition, and we need to consider multiple CP factors. This can be computed by performing single-factor CP decomposition sequentially to the residuals remaining after subtracting out the previously computed factors. This is called Tensor Power Method; detailed algorithm is as follows:

Algorithm 1: Multiple CP Factors

initialize $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$
for $k = 1, \cdots, K$ **do**
  repeat
    $\mathbf{u_k} \leftarrow \frac{\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k}{||\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k||_F}$
    $\mathbf{v_k} \leftarrow \frac{\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k}{||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k||_F}$
    $\mathbf{w_k} \leftarrow \frac{\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k}{||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k||_F}$
  until *converges*;
  $d_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k$
  $\hat{\boldsymbol{\mathcal{X}}} \leftarrow \hat{\boldsymbol{\mathcal{X}}} - d_k \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$
end

2.4 Sparse CP Decomposition

Now, we need to incorporate sparsity into CP decomposition. The author solved this problem by considering L_1-norm regularization term.

2.4.1 Problems in Sparse CP Decomposition

The new single-factor Sparse CP decomposition is as follows:

$$\text{maximize}_{\mathbf{u},\mathbf{v},\mathbf{w}} \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}}||\mathbf{u}||_1 - \rho_{\mathbf{v}}||\mathbf{v}||_1 - \rho_{\mathbf{w}}||\mathbf{w}||_1$$

$$\text{subject to } \mathbf{u}^T\mathbf{u} \leq 1, \mathbf{v}^T\mathbf{v} \leq 1, \mathbf{w}^T\mathbf{w} \leq 1 \tag{2}$$

Here $\rho_{\mathbf{u}}, \rho_{\mathbf{v}}$ and $\rho_{\mathbf{w}}$ are non-negative bandwidth parameters controlling the amount of sparsity in the tensor factors and in (2) the constraints are inequalities. By relaxing the constraints the optimization problem

becomes simpler. The solution of this problem is as follows:

$$\hat{\mathbf{u}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})}{||S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})||_F}, & \text{if } ||S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{v}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}})}{||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}})||_F}, & \text{if } ||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{w}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}})}{||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}})||_F}, & \text{if } ||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $S(\cdot, \rho)$ is the soft-thresholding operator: $S(\cdot, \rho) = \text{sign}(\cdot)(|\cdot| - \rho)_+$

When optimizing the problem (2) with respect to $\mathbf{u}$, the KKT conditions imply that $\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}} \Gamma(\mathbf{u}^*) - 2\gamma^* \mathbf{u}^* = 0$ and $\gamma^*((\mathbf{u}^*)^T \mathbf{u}^* - 1) = 0$ where $\Gamma(\mathbf{u})$ is the subgradient of $||\mathbf{u}||_1$, and γ is a Lagrange multiplier. Detailed derivation is given in the Appendix section.

The multiple sparse CP factors can be obtained likewise in Algorithm 1 in the Section 2.3.

Algorithm 2: Multiple Sparse CP Factors

initialize $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$
for $k = 1, \cdots, K$ **do**
  repeat
    $\mathbf{u}_k \leftarrow \begin{cases} \frac{S(\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{u}})}{||S(\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{u}})||_F}, & \text{if } ||S(\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{u}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$
    $\mathbf{v}_k \leftarrow \begin{cases} \frac{S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{v}})}{||S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{v}})||_F}, & \text{if } ||S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k, \rho_{\mathbf{v}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$
    $\mathbf{w}_k \leftarrow \begin{cases} \frac{S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k, \rho_{\mathbf{w}})}{||S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k, \rho_{\mathbf{w}})||_F}, & \text{if } ||S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k, \rho_{\mathbf{w}})||_F > 0 \\ 0, & \text{otherwise} \end{cases}$
  until *converges*;
  $d_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k$
  $\hat{\boldsymbol{\mathcal{X}}} \leftarrow \hat{\boldsymbol{\mathcal{X}}} - d_k \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$
end

2.5 Amount of Variance Explained

It is necessary to check the amount of variance explained by k PCs when it comes to PCA literature so that the number of factors, k, can be determined. Thus, it is also required to obtain the amount of variance explained with sparse higher-order PCA framework in order to determine the extent of dimension reduction achieved. With matrix data, it is possible to recover the data matrix exactly with PCs; however, with higher-order tensor, it is not true. This is because unlikely the matrix data, it is not possible to write common tensor factorization with orthonormal basis vectors, diagonal tensor core and mode multiplication. This implies the tensor data cannot be recovered exactly. Thus d_k^2 does not give the information regarding proportion of variance explained by the tensor decomposition any longer.

Thus, we need a different approach. We know that the cumulative proportion of variance explained by obtaining the ratio of the variance of the data projected onto the first k singular vectors to the variance of the original data. So, this idea can also be easily extended into higher-order tensor decomposition. Let $\mathbf{P}_k^{(U)}, \mathbf{P}_k^{(V)}$ and $\mathbf{P}_k^{(W)}$ be projection matrices with exactly k eigenvalues equal to one. Thus, $\mathbf{P}_k^{(U)} = \mathbf{U}_k(\mathbf{U}_k^T\mathbf{U}_k)^{-1}\mathbf{U}_k^T$ where $\mathbf{U}_k = [\mathbf{u}_1, \cdots, \mathbf{u}_k]$. Then the cumulative proportion of variance explained by the first k higher-order PC's is as follows:

$$\frac{||\boldsymbol{\mathcal{X}} \times_1 \mathbf{P}_k^{(U)} \times_2 \mathbf{P}_k^{(V)} \times_3 \mathbf{P}_k^{(W)}||_F^2}{||\boldsymbol{\mathcal{X}}||_F^2}$$

It is notable that eigenvalues of the covariance matrix were not used in this calculation.

So far, the author dealt with the case of L_1-norm regularization term; however, other regularization terms can be used as well.

3 Summary

Sparsity in higher-order PCA is desirable with several reasons, and by the simple and novel algorithms introduced in this paper, it is now possible to compute the sparse HOPCA. Instead of exploiting Tucker's decomposition (HOSVD), HOPCA can be achieved by CP decomposition with tensor power method. Compared to HOSVD, Sparse CP decomposition was both mathematically and computationally appealing. Furthermore, CP decomposition can also be done by Alternating Least Squares algorithm (ALS). As a further research, the CP decomposition performances by TPM and ALS should be compared in many aspects such as computing time, convergence rate, or amount of variance explained.

4 Citations

1. Allen, G. I. (2012). Sparse Higher-Order Principal Components Analysis. In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics

2. Kolda, T. and B. Bader (2009). Tensor decompositions and applications. SIAM review 51 (3), 455–500.

3. Harshman, R. (1970). Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. UCLA Working Papers in Phonetics.

4. Tucker, L. (1966). Some mathematical notes on threemode factor analysis. Psychometrika 31 (3), 279–311.

5. Kiers, H. A. L. (2000). Towards a standardized notation and terminology in multiway analysis. J. Chemometrics, 14, pp. 105–122.

Appendix

CANDECOMP/PARAFAC Decomposition

The CANDECOMP (canonical decomposition) and the PARAFAC (parallel factors) were introduced in 1970 to the psychometrics community. And in 2000, Kier (2000) discovered CANDECOMP/PARAFAC (CP) decomposition by combining these two decomposition methods. The CP decomposition factorizes a tensor into a sum of component rank-one tensors. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q}$, then we have:

$$\boldsymbol{\mathcal{X}} \approx \sum_{i=1}^{k} \mathbf{u}_i \circ \mathbf{v}_i \circ \mathbf{w}_i$$

where k is a positive integer and $\mathbf{u}_i \in \mathbb{R}^n$, $\mathbf{v}_i \in \mathbb{R}^p$ and $\mathbf{w}_i \in \mathbb{R}^q$ for $i = 1, \cdots, k$. And with an assumption that $\mathbf{u}_i$, $\mathbf{v}_i$ and $\mathbf{w}_i$ are length one vectors, one can rewrite it as follows:

$$\boldsymbol{\mathcal{X}} \approx \sum_{i=1}^{k} d_i \mathbf{u}_i \circ \mathbf{v}_i \circ \mathbf{w}_i$$

where $d_i \in \mathbb{R}$. One interesting property of higher-order tensors is their rank decompositions are often unique whereas matrix decompositions are not.

Derivation of Equation (2)

The Lagrangian is given by:

$$\nabla_{\mathbf{u}} L(\mathbf{u}, \gamma) = \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}} \Gamma(\mathbf{u}) - 2\gamma \mathbf{u}$$

$$\text{Let } \mathbf{u}^* \text{ be } \mathbf{u} \text{ such that } \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}} \Gamma(\mathbf{u}^*) - 2\gamma^* \mathbf{u}^* = 0$$

With L_1 norm, $\Gamma(\mathbf{u}) = \text{sign}(\mathbf{u})$, then we can rewrite the equation above as follows:

$$\mathbf{u}^* = \frac{\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \text{sign}(\mathbf{u}^*)\rho_{\mathbf{u}}}{2\gamma^*}$$

If $\mathbf{u}^* < 0$, then $\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} < -\rho_{\mathbf{u}}$, and if $\mathbf{u}^* > 0$, then $\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} > \rho_{\mathbf{u}}$. Therefore, $|\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}| > \rho_{\mathbf{u}}$; this implies $\text{sign}(\mathbf{u}^*) = \text{sign}(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w})$. So we have,

$$\begin{aligned} \mathbf{u}^* &= \frac{\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \text{sign}(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w})\rho_{\mathbf{u}}}{2\gamma^*} \\ &= \frac{S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w},\ \rho_{\mathbf{u}})}{2\gamma^*} \end{aligned}$$

We can obtain γ^* from $\partial L(\mathbf{u}, \gamma)/\partial\gamma|_{\mathbf{u}=\mathbf{u}^*} = (\mathbf{u}^*)^T\mathbf{u}^* - 1 = 0$. This implies that $\mathbf{u}^*$ is a unit vector; therefore:

$$\mathbf{u}^* = \frac{S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w},\ \rho_{\mathbf{u}})}{||S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w},\ \rho_{\mathbf{u}})||_F}$$