Sparseness Constrained Tensor Factorization Algorithm for Dictionary Learning over High Dimensional Space

By Syed Zubair[1], Wei Dai[2], and Wenwu Wang[1]

[1]Department of Electronic Engineering, University of Surrey, UK
[2] Department of Electrical and Electronic Engineering, Imperial College London, UK
Emails: s.zubair@surrey.ac.uk; wei.dai1@imperial.ac.uk; w.wang@surrey.ac.uk

Abstract

Dictionary learning algorithms are typically derived for dealing with one or two dimensional signals using vector-matrix operations. Little attention has been paid to the problem of dictionary learning over high dimensional tensor data. We propose a new algorithm for dictionary learning based on tensor factorization using a Tucker model. In this algorithm, sparseness constraints are applied to the core tensor, of which the n-mode factors are learned from the input data in an alternate minimization manner using gradient descent. Simulations are provided to show the convergence property and the reconstruction performance of the proposed algorithm.

1. Introduction

Learning signal features and structures is important for obtaining its succinct representation that can be used for various applications such as source separation and signal classification. Dictionary learning algorithms have recently been used for learning such representations as given in Aharon & Elad (2006) . However, these algorithms are mostly limited to one or two dimensional signals. With content rich applications emerging nowadays, signal dimensionality is constantly increasing e.g. in video signals. Moreover, a low-dimensional signal such as an audio signal, can be cast in a higher dimensional space, e.g. in a space-time-frequency domain. Hence, it becomes highly desirable if those algorithms can deal with higher dimensional data, such as tensor data.

Tensor factorization and decomposition have recently attracted attention in signal processing community, for processing high dimensional signals. PARAFAC and TUCKER decompositions are two such algorithms. PARAFAC decomposes the tensor as a sum of k rank-1 tensors while the TUCKER method computes the orthonormal subspaces corresponding to each mode of the tensor. This can be treated as higher order principal component analysis. However, these methods do not explicitly enforce signal sparsity despite its benefits in signal representations in various applications.

In this paper, we propose sparseness constrained tensor factorization algorithm for dictionary learning over high-dimensional space. The proposed algorithm, similar to standard dictionary learning algorithms, is also a two-stage iterative process: sparse coding and dictionary update. First, given an initial dictionary tensor, we extend orthogonal matching pursuit (OMP) to find the sparse coefficient tensor. This is done by first converting the input tensor into a vector form which can further be represented in terms of the product of dictionary matrix and a coefficient vector, where the dictionary matrix is equivalent to the Kronecker product of the dictionary matrices associated with each mode of the input tensor, and the coefficient vector is the vectorized form of the coefficient tensor. Then the dictionaries corresponding to each mode are updated, by using gradient descent method. For instance, in case of a three dimensional tensor, its learned dictionary is comprised of Kronecker product of three dictionary matrices corresponding to each mode, which can be updated using the gradient descent, by fixing two of the three matrices and updating the remaining one. This is repeated for all the matrices corresponding to each mode. The resulting coefficient vector and dictionary matrices are again folded back to obtain the TUCKER representation of the input signal tensor.

2. The Proposed Algorithm

A signal of a high dimension is considered as a tensor. Here for simplicity, we consider $\underline{\mathrm{Y}}$ as a tensor of three dimensions e.g. $\underline{\mathrm{Y}} \in R^{I_1 \times I_2 \times I_3}$. It can also be called as a three-way signal. Matrix is a form of two-way signal and vector is considered one-way signal. A tensor can be unfolded to an n-mode matrix form and represented as $\mathrm{Y}_{(n)}$. For a three-way tensor, the mode-n matrix can be extracted by changing all the indices in the tensor except the n-th index. Hence a three-way tensor can be unfolded into any of its mode-n matrix. For example, the mode-1 unfolded matrix of tensor $\underline{\mathrm{Y}}$, i.e. $\mathrm{Y}_{(1)}$, has a dimension $R^{I_1 \times I_2 I_3}$. Similarly the mode-2 unfolded matrix $\mathrm{Y}_{(2)}$ has a dimension $R^{I_2 \times I_1 I_3}$. Tensor decomposition can be formulated as:

$$\underline{\mathrm{Y}} = \underline{\mathrm{X}} \times_1 \mathrm{A} \times_2 \mathrm{B} \times_3 \mathrm{C} = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \sum_{m_3=1}^{M_3} x_{m_1 m_2 m_3} \mathbf{a}_{m_1} \circ \mathbf{b}_{m_2} \circ \mathbf{c}_{m_3} \tag{2.1}$$

where $\circ$ is the outer product between the vectors. $\mathrm{A} \in R^{I_1 \times M_1}$, $\mathrm{B} \in R^{I_2 \times M_2}$ and $\mathrm{C} \in R^{I_3 \times M_3}$ are orthogonal factor matrices composed of **a**, **b** and **c** vectros and can be considered as principal components along each mode of the tensor. $\underline{\mathrm{X}} \in R^{M_1 \times M_2 \times M_3}$ is a core tensor. This form of decomposition was suggested by Tucker (1966), hence it is called Tucker decomposition. It can be represented element-wise as

$$y_{i_1 i_2 i_3} = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \sum_{m_3=1}^{M_3} x_{m_1 m_2 m_3} \; a_{i_1 m_1} \; b_{i_2 m_2} \; c_{i_3 m_3} \tag{2.2}$$
$$for \;\; i_1 = 1, \ldots, I_1, i_2 = 1, \ldots, I_2, i_3 = 1, \ldots, I_3$$

If the core tensor $\underline{\mathrm{X}}$ is super-diagonal and $M_1 = M_2 = M_3$, then this can be considered as PARAFAC decomposition introduced by Harshman (1970).

To learn Tucker decomposition with a sparsity constraint on the core tensor $\underline{\mathrm{X}}$, our objective function in (2.1) takes the form:

$$\min_{\underline{\mathrm{X}},\mathrm{A},\mathrm{B},\mathrm{C}} \| \underline{\mathrm{Y}} - \underline{\mathrm{X}} \times_1 \mathrm{A} \times_2 \mathrm{B} \times_3 \mathrm{C} \| \quad s.t. \;\; x_{m_1 m_2 m_3} = 0 \; \forall \; (m_1, m_2, m_3) \notin \mathcal{M}_1 \times \mathcal{M}_2 \times \mathcal{M}_3 \tag{2.3}$$

where $\mathcal{M}_n = [m_n^1, m_n^2, \ldots, m_n^{s_n}]$ denotes the subset of indices of non-zero values in the core tensor for mode n ($n = 1, 2, 3$), and s_n represents the n-mode sparsity, showing the number of selected columns of each factor required for the Tucker representation. The total sparsity of three way core tensor is denoted by $s = s_1 \times s_2 \times s_3$. Here we assume that the size of the core tensor $\underline{\mathrm{X}}$ is larger than or equal to the size of $\underline{\mathrm{Y}}$ ($M_n \geq I_n$).

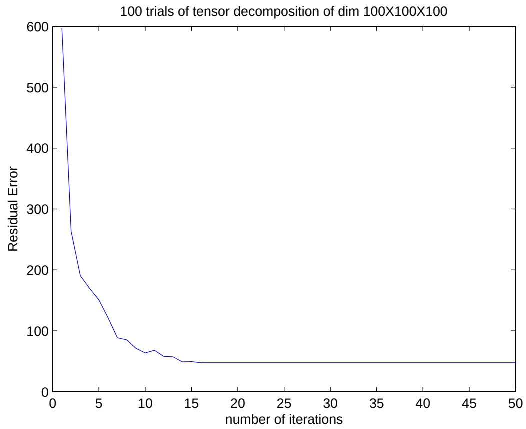


FIGURE 1. Convergence of GradTensor over 100 trials.

The Tucker decomposition factors and the core tensors are computed in a two-step process. In the first step, the sparse core tensor is computed using the tensor version of orthogonal matching pursuit (Tensor-OMP) suggested by Caiafa & Cichocki (2012) with Tucker factors initialized by M_n, leading to left singular values of mode-n matrices of input tensor $\underline{\mathrm{Y}}$. Once the sparse core tensor is obtained, the Tucker factors are computed by gradient descent with their gradients calculated as follows:

$$\begin{aligned}\nabla \mathrm{A} &= (\mathrm{Y}_{(1)} - \mathrm{A}\mathrm{X}_{(1)}(\mathrm{C} \otimes \mathrm{B}))(\mathrm{C} \otimes \mathrm{B})^{\dagger} \\ \nabla \mathrm{B} &= (\mathrm{Y}_{(2)} - \mathrm{B}\mathrm{X}_{(2)}(\mathrm{C} \otimes \mathrm{A}))(\mathrm{C} \otimes \mathrm{A})^{\dagger} \\ \nabla \mathrm{C} &= (\mathrm{Y}_{(3)} - \mathrm{C}\mathrm{X}_{(3)}(\mathrm{B} \otimes \mathrm{A}))(\mathrm{B} \otimes \mathrm{A})^{\dagger}\end{aligned} \tag{2.4}$$

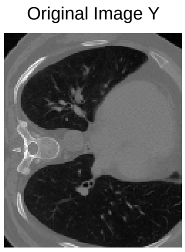
where $\otimes$ is the Kronecker product. The Tucker components are learned in an alternate manner such that when learning one component like A, all the other components and the core tensor are held fixed. In this way, all the Tucker components are learned. This two-stage learning process alternates between factors decomposition and sparse core tensor learning until a stopping criterion is reached.

These gradients are used to update the Tucker factors one after another. After all the factors are updated, Tensor-OMP is used to obtain the sparse core tensor.

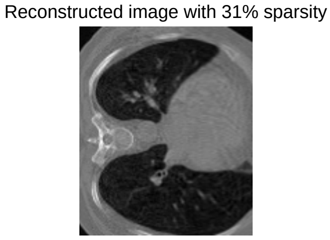
3. Experiments and Results

In the first experiment, we test the convergence of our proposed method by applying it on a synthetically generated tensor of size $100 \times 100 \times 100$. The tensor is generated from the mode factors and the sparse core tensor whose elements are Gaussian distributed. The sparse core tensor has a fixed mode sparsity of $\mu = s_n/M_n = 1/6$. The algorithm convergence averaged over 100 independent trials, is shown in Figure 1.

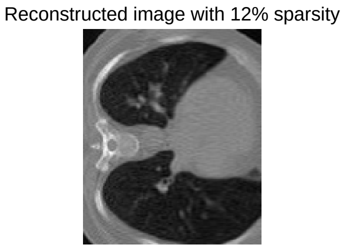
For the second experiment, we learn high dimensional signal features for human abdomen 3-D image of size $151 \times 125 \times 141$ by our proposed algorithm. This dataset is given by Vandemeulebroucke et al (2007). We set $M_n = 1.5I_n$ and the fixed mode sparsity as $\mu = 1/2.2$ and $\mu = 1/3$ respectively, which is equal to the total sparsity level of the core tensor at 31% and 12%, respectively. The 50th slice of the image is reconstructed by the learned sparse core tensor and the Tucker factors, as shown in Figure 2, where the original image slice can be compared with the reconstructed slices using the two different levels



(a) Original image of the human abdomen.



(b) Reconstructed image with the core tensor sparsity of 31%.



(c) Reconstructed image with the core tensor sparsity of 12%.

FIGURE 2. Comparison between the original image and the reconstructed images using two different sparsity levels of the core tensor.

of sparsity. It can be observed that the reconstructed image using the atoms learned by the proposed sparse tensor learning algorithm resembles the original image very nicely. In the future, we will test the performance of the proposed algorithm more thoroughly. We will also consider the application of the proposed method for the problems of signal classification and source separation.

REFERENCES

AHARON, M & ELAD, M. & BRUCKSTEIN, A. 2006 K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing* **54**, 4311-4322.

CAIAFA, C. F. & CICHOCKI, A. 2012 Block sparse representation of tensors using Kronecker bases. *IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, 2709-2712.

HARSHMAN, R. A. 1970 Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA working papers in phonetics* **16**, 1-84.

TUCKER, L. R. 1966 Some mathematical notes on 3-mode factor analysis. *Psychometrika* **31**, 279–311.

KOLDA, T. G. & BADER, B. W. 2009 Tensor decompositions and applications. *SIAM Review* **51**, 455–500.

VANDEMEULEBROUCKE, J. & SARRUT, D. & CLARYSSE, P. 2007 The POPI-model, a point-validated pixel-based breathing thorax model *In Proceeding of the XVth ICCR Conference.*