DISCRIMINATIVE TENSOR DICTIONARIES AND SPARSITY FOR SPEAKER IDENTIFICATION

S. Zubair, W. Wang

Centre for Vision, Speech and Signal Processing
University of Surrey, UK

J. A. Chambers

School of Electronic, Electrical and Systems Engineering
Loughborough University, UK

ABSTRACT

Dictionary learning algorithms based upon matrices/vectors have been used for signal classification by incorporating different constraints such as sparsity, discrimination promoting terms or by learning a classifier along with the dictionary. However, because of the limitations of matrix based dictionary learning algorithms in capturing the underlying subspaces of the data presented in the literature, we learn tensor dictionaries with discriminative constraints and extract classifiers out of the dictionaries learned over each mode of the tensor. This algorithm, named as GT-D, is then used for the speaker identification. We compare classification performance of our proposed algorithm with other state-of-the-art tensor decomposition algorithms for the speaker identification problem. Our results show the supremacy of our proposed method over other approaches.

Index Terms— Tensor Factorization, Sparse Representations, Classification, Dictionary Learning

1. INTRODUCTION

Learning the features and structures of a signal is important for obtaining a succinct representation that can be used for various applications such as source separation and signal classification. Dictionary learning algorithms emerging from sparse representations have recently been used for learning such representations as given in [1]. 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. This preserves the structure of the signal which may otherwise be lost when used in a low dimensional form. Hence, it becomes highly desirable for dictionary learning algorithms to be able to learn signal features from higher dimensional data, such as tensor data.

Tensor factorization and decomposition have recently attracted attention in the signal processing community, for processing high dimensional signals. PARAFAC [2] and TUCKER [3] decompositions are two such classical 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 for various applications.

Tensor decomposition and dictionary learning algorithms have also been used for signal classification. Inspired by the non-negative matrix factorization (NMF) techniques due to Lee and Seung [4], E. Benetos et al. in [5] introduced non-negative PARAFAC decomposition with multiplicative updates and designed a classifier from learned PARAFAC factors for the classification of musical genres. Another non-negative tensor decomposition [6] used the TUCKER model to learn the dictionaries over its each mode and used its core tensor as features to supply to a conventional classifier such as a SVM. Similarly, non-negative versions of TUCKER decomposition have also been proposed in [7] and [8].

In this paper, we propose a discriminative tensor dictionary learning algorithm based on the TUCKER model with sparsity constraints over its core tensor. The discriminative constraint is applied by incorporating the Fisher criterion while learning the tensor dictionaries. The sparse core tensor is calculated by a tensor extended version of the greedy algorithm, Tensor Orthogonal Matching Pursuit (TOMP) [9].

The organization of the whole paper is as follows: Section 2 formulates an objective function for the tensor dictionary learning problem. Section 3 presents the proposed discriminative dictionary learning method for high dimensional data, GT-D. Section 4 shows experiments along with their results and Section 5 concludes the paper.

2. PROBLEM FORMULATION AND OPTIMIZATION CRITERION

We consider a third-order tensor $\underline{\mathbf{Y}} \in R^{I_1 \times I_2 \times I_3}$ which can be represented in terms of its factors using the TUCKER

978-1-4799-3109-5/14/$31.00 ©2014 IEEE

model as:

$$\begin{aligned}\underline{\mathrm{Y}} &= \underline{\mathrm{X}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \times_3 \mathrm{A}^{(3)} \\ &= \sum_{m_1=1}^{M_1}\sum_{m_2=1}^{M_2}\sum_{m_3=1}^{M_3} x_{m_1 m_2 m_3}\mathbf{a}_{m_1}^{(1)} \circ \mathbf{a}_{m_2}^{(2)} \circ \mathbf{a}_{m_3}^{(3)}\end{aligned} \quad (1)$$

where $\circ$ is the outer product between the vectors. $\mathrm{A}^{(1)} \in R^{I_1 \times M_1}$, $\mathrm{A}^{(2)} \in R^{I_2 \times M_2}$ and $\mathrm{A}^{(3)} \in R^{I_3 \times M_3}$ are n-mode dictionaries composed of $\mathbf{a}^{(n)}$ vectors for $n = 1, 2, 3$. $\underline{\mathrm{X}} \in R^{M_1 \times M_2 \times M_3}$ is a core tensor. This form of decomposition was suggested in [3], hence it is called a TUCKER decomposition. A tensor can be unfolded to a mode-n 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 matrices. For example, the mode-1 unfolded matrix of tensor $\underline{\mathrm{Y}}$, i.e. $\mathrm{Y}_{(1)}$, has a dimension $R^{I_1 \times I_2 I_3}$

To learn discriminative dictionaries, the Fisher criterion on the n-mode dictionaries $\mathrm{A}^{(n)}$ is applied as:

$$max\ J(\mathrm{A}^{(1)}, \mathrm{A}^{(2)}, \mathrm{A}^{(3)}) = \frac{\Psi_b(\underline{\mathrm{X}})}{\Psi_w(\underline{\mathrm{X}})} \quad (2)$$

where Ψ_w and Ψ_b are tensor based within class and between class scatter terms respectively, given by $\Psi_w = tr(\mathrm{A}^{(n)T}\mathrm{S}_w^{(n)}\mathrm{A}^{(n)})$ and $\Psi_b = tr(\mathrm{A}^{(n)T}\mathrm{S}_b^{(n)}\mathrm{A}^{(n)})$ and $tr(\cdot)$ is the trace of a matrix, $(\cdot)^T$ denotes the transpose operation, $\mathrm{S}_w^{(n)}$ and $\mathrm{S}_b^{(n)}$ are the n-mode within-class and between class matrices computed by:

$$\mathrm{S}_w^{(n)} = \sum_{c=1}^{C}\sum_{i=1}^{K_c}\left(\mathrm{Y}_{i(n)}^c - \mathrm{Y}_{\mu(n)}^c\right)\tilde{\mathrm{A}}_{-n} \times \tilde{\mathrm{A}}_{-n}^T\left(\mathrm{Y}_{i(n)}^c - \mathrm{Y}_{\mu(n)}^c\right)^T \quad (3)$$

$$\mathrm{S}_b^{(n)} = \sum_{c=1}^{C}\left(\mathrm{Y}_{\mu(n)}^c - \mathrm{Y}_{\mu(n)}\right)\tilde{\mathrm{A}}_{-n} \times \tilde{\mathrm{A}}_{-n}^T\left(\mathrm{Y}_{\mu(n)}^c - \mathrm{Y}_{\mu(n)}\right)^T \quad (4)$$

where C is the total number of classes, K_c is the number of training samples in each class c, $\tilde{\mathrm{A}}_{-n} = \mathrm{A}^{(1)} \times_1 \cdots \times_{n-1} \mathrm{A}^{(n-1)} \times_{n+1} \mathrm{A}^{(n+1)} \cdots \times_N \mathrm{A}^{(N)}$, $\mathrm{Y}_{\mu(n)}^c$ and $\mathrm{Y}_{\mu(n)}$ are the n-mode unfolded forms of class-mean and overall mean of the input tensors, respectively.

Let $\tilde{\mathrm{A}} = \mathrm{A}^{(1)} \times_1 \mathrm{A}^{(2)} \times_3 \mathrm{A}^{(3)}$, then to learn the discriminative tensor dictionaries with a sparsity constraint on the core tensor $\underline{\mathrm{X}}$, our objective function for model (1) takes the form:

$$\begin{aligned}\mathcal{F}_D(\underline{\mathrm{X}}, \tilde{\mathrm{A}}) = \min_{\underline{\mathrm{X}}, \tilde{\mathrm{A}}} \| \underline{\mathrm{Y}} - \underline{\mathrm{X}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \times_3 \mathrm{A}^{(3)} \|_F^2 \\ +\lambda_1\Psi_w - \lambda_2\Psi_b \\ s.t. \quad \| \underline{\mathrm{X}} \|_0 \leq s\end{aligned} \quad (5)$$

where $\| \cdot \|_F$ is the Frobenius norm, $\| \cdot \|_0$ is the ℓ_0 norm which counts the number of non-zeros in the core tensor $\underline{\mathrm{X}}$, λ_1 and λ_2 are the penalty parameters and the total sparsity (i.e. the number of non-zeros) of the three way core tensor is denoted by $s = s_1 \times s_2 \times s_3$ where s_n represents the n-mode sparsity, showing the number of selected columns of each n-mode dictionary required for the TUCKER representation. To calculate the sparse core tensor $\underline{\mathrm{X}}$, we use a greedy algorithm TOMP proposed in [9].

3. GRADTENSOR FOR DISCRIMINATIVE DICTIONARIES (GT-D)

The tensor dictionaries and the core tensor are computed in a two-step process. In the first step, the sparse core tensor is computed using TOMP with tensor dictionaries initialized by M_n left leading singular vectors of the n-mode unfolded matrices of the input tensor $\underline{\mathrm{Y}}$. Once the sparse core tensor is obtained, in the next step, the discriminative tensor dictionaries are learned iteratively by gradient descent in an alternating manner.

Mathematically, equation (1) can be represented in an unfolded form as

$$\begin{aligned}\mathrm{Y}_{(1)} &= \mathrm{A}^{(1)}\mathrm{X}_{(1)}(\mathrm{A}^{(3)} \otimes \mathrm{A}^{1})^T \\ \mathrm{Y}_{(2)} &= \mathrm{A}^{(2)}\mathrm{X}_{(2)}(\mathrm{A}^{(3)} \otimes \mathrm{A}^{(1)})^T \\ \mathrm{Y}_{(3)} &= \mathrm{A}^{(3)}\mathrm{X}_{(3)}(\mathrm{A}^{(2)} \otimes \mathrm{A}^{(1)})^T\end{aligned} \quad (6)$$

where $\otimes$ denotes the Kronecker product. To calculate the n-mode dictionary $\mathrm{A}^{(n)}$ in the unfolded form, the minimization of equation (5) can be written as

$$\begin{aligned}\arg\min_{\mathrm{A}^{(n)}} \| \mathrm{Y}_{(n)} - \mathrm{A}^{(n)}\mathrm{X}_{(n)}(\tilde{\mathrm{A}}_{-n})^T \|_F^2 + \\ \lambda_1 tr(\mathrm{A}^{(n)T}\mathrm{S}_w^{(n)}\mathrm{A}^{(n)}) - \lambda_2 tr(\mathrm{A}^{(n)T}\mathrm{S}_b^{(n)}\mathrm{A}^{(n)})\end{aligned} \quad (7)$$

where in this case, $n = 1, 2, 3$. The gradient of (7) with respect to $\mathrm{A}^{(n)}$ can be calculated by

$$\begin{aligned}\nabla\mathcal{F}_D(\mathrm{A}^{(n)}) = (\mathrm{Y}_{(n)} - \mathrm{A}^{(n)}\mathrm{X}_{(n)}\tilde{\mathrm{A}}_{-n})\left\{\mathrm{X}_{(n)}\tilde{\mathrm{A}}_{-n}^T\right\}^\dagger + \\ 2\lambda_1\mathrm{S}_w^{(n)}\mathrm{A}^{(n)} - 2\lambda_2\mathrm{S}_b^{(n)}\mathrm{A}^{(n)}\end{aligned} \quad (8)$$

where $\dagger$ is the pseudo-inverse of the matrix. The update for the n-mode discriminative dictionary is

$$\mathrm{A}^{(n)(k+1)} = \mathrm{A}^{(n)(k)} - \gamma_1\nabla\mathcal{F}_D^{(k)}(\mathrm{A}^{(n)}) \quad (9)$$

where γ_1 is the step size and k is the current step of the gradient descent algorithm. These discriminative tensor dictionaries are learned in an alternate minimization manner such that when learning one dictionary such as $\mathrm{A}^{(1)}$, all the other dictionaries and the core tensor are held fixed. In this way,

all the dictionaries are updated. In the next iteration, these learned dictionaries are used to find out the sparse core tensor in the sparse coding stage. This two-stage learning process alternates between tensor dictionaries learning and sparse core tensor update until a stopping criterion is reached. Algorithm 1 gives the summary of the whole algorithm.

As these dictionaries are learned by the gradient descent, the minimization of the objective function may lead to local minima. To improve convergence, all the dictionaries are initialized by the left leading factors of the input tensor data. Though we don't have an explicit proof for the convergence of the algorithm yet, the simulations on synthetic data show good convergence of the algorithm as confirmed in the next section.

Algorithm 1: GradTensor for discriminative dictionaries (GT-D)

Task: Find n-mode dictionaries $\mathrm{A}^{(n)} \in R^{I_n \times M_n}$ and sparse core tensor $\underline{\mathrm{X}} \in R^{M_1 \times M_2 \times M_3}$ that give discriminative dictionaries and sparsest representation of input signal tensor $\underline{\mathrm{Y}} \in R^{I_1 \times I_2 \times I_3}$ with predefined sparsity $s = s_1 \times s_2 \times s_3$.
Require: Input signal $\underline{\mathrm{Y}}$, sparse core tensor $\underline{\mathrm{X}}$, maximum sparsity value (total number of non-zeros) s, step size γ_1, tolerance parameters ϵ_1 and ϵ_2,
Output: All n-mode dictionaries $\mathrm{A}^{(n)}$.
Initialization: Each mode-n dictionary $\mathrm{A}^{(n)}$ is initialized by M_n left leading vectors of Y_n, where $n = 1, 2, 3$ is the index of the modes of a tensor.
Repeat until convergence:(i.e. $\mathcal{F}_D(\underline{\mathrm{X}}, \tilde{\mathrm{A}}) \leq \epsilon_2$)

1. *Sparse Coding Stage*: Use TOMP to find sparse core tensor $\underline{\mathrm{X}}$.
2. *Dictionary Learning Stage*: Learn discriminative dictionaries $\mathrm{A}^{(n)}$ for each mode by the discriminative gradient descent.

   for $n = 1, 2, 3$

   - Find n-mode scatter matrices $\mathrm{S}_w^{(n)}$ and $\mathrm{S}_b^{(n)}$ via equation (3) and (4).
   - Calculate gradient for $\mathrm{A}^{(n)}$, $\nabla \mathcal{F}_D(A^{(n)})$. While fixing all the other dictionaries and sparse core tensor, update discriminative $\mathrm{A}^{(n)}$ by (8) and (9) until the error between two consecutive iterations for $\mathrm{A}^{(n)}$ update reaches ϵ_1 or the number of iterations reaches its maximum limit.

   end

4. EXPERIMENTS AND RESULTS

To investigate the classification performance of our proposed algorithm, GT-D, over other tensor decomposition methods, we apply our GT-D on a speaker identification problem and compare the results with baseline TUCKER decomposition algorithms such as standard TUCKER method TALS [3], and the state-of-the-art GT-G [10], HALS [6], TCCD [11] and APG [12] methods. We use all of these algorithm with their default parameters. However, the size of their dictionaries and the sparsity of their core tensors (where applicable) are set to the same size and value as that of the GT-D. To show the benefit of tensor based methods over matrix based methods, we also compare our proposed algorithm with the K-SVD [1] dictionary learning algorithm.

4.1. Signal Classification

Signal classification is performed by projecting the feature matrix of the test signals onto the tensor extracted from the learned factors of the tensors. For a given input tensor, the classifier in each case of the competing algorithms is computed by

$$\underline{\mathrm{D}} = \underline{\mathrm{X}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \tag{10}$$

where $\underline{\mathrm{D}}$ is the learned basis tensor. This basis tensor $\underline{\mathrm{D}}$ is used to classify the test feature matrix and is determined during the training phase. By following the equation (6), the input signal $\underline{\mathrm{Y}} \in R^{I_1 \times I_2 \times I_3}$ can be represented in terms of the learned classifier tensor in a mode-1 unfolded form as

$$\mathrm{Y}_{(1)} = \mathrm{I}_{\mathrm{A}^{(1)}} \mathrm{D}_{(1)} (\mathrm{A}^{(3)} \otimes \mathrm{I}_{\mathrm{A}^{(2)}})^T \tag{11}$$

where $\mathrm{I}_{\mathrm{A}^{(1)}}$ and $\mathrm{I}_{\mathrm{A}^{(2)}}$ are the identity matrices of the same size as $\mathrm{A}^{(1)}$ and $\mathrm{A}^{(2)}$, respectively. For each classification application, the test feature matrix Y^{test} is projected on to each class-specific classifier tensor obtained by (10) to get the coefficients by using the least squares method. The resulting coefficients are used to reconstruct the test feature matrix $\hat{\mathrm{Y}}_i^{test}$ with the help of each $\mathrm{D}_{(1)_i}$. The class label of the basis tensor that gives the minimum residual error in signal reconstruction is the predicted label of the test signal. Mathematically,

$$l = arg \ \min_i \left(\mathrm{Y}^{test} - \hat{\mathrm{Y}}_i^{test} \right) \tag{12}$$

where l is the predicted label of the test signal matrix Y^{test}. For this classification problem, a subset of the TIMIT [13] corpus is selected for speaker identification of 5 speakers with 10 utterances (sentences) per speaker, resulting in a total of 50 utterances. Each utterance (sound file) has a different duration with an average value of 1.5 seconds. All the sound files are sampled at a sampling frequency of 16 kHz. Each sound file's (.wav format) raw data is first decomposed into frames of 1000 samples and each frame is converted to 13-dimensional Linear Predictive Coding (LPC) features with 50% overlap. Hence, each utterance is converted to a number of LPC feature vectors which are stacked one after the other to make a matrix of size $I_1 \times I_2 = 13 \times 99$. Since there are 10 utterances per class (speaker), hence all LPC matrices

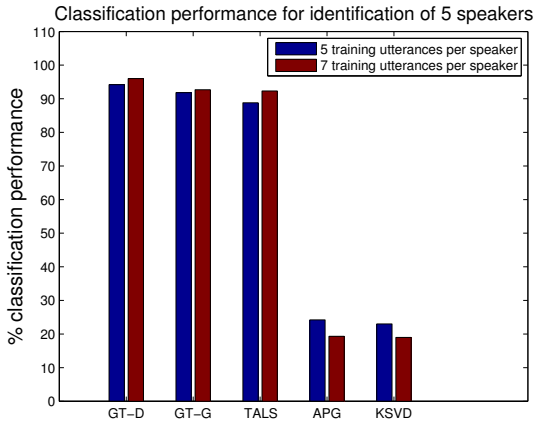

Fig. 1. Classification performances for the identification of 5 speakers for different decomposition algorithms with different number of utterances per speaker.

corresponding to all utterances per class are stacked one after the other to form a tensor of size $I_1 \times I_2 \times I_3 = 13 \times 99 \times 10$. In this way, each frontal slice of a tensor represents one utterance per class (speaker). From each class tensor, we randomly select training and testing signals by selecting frontal slices. For 5-speakers, we obtain 5 tensors with each having the same size.

We perform two experiments, with 5 and 7 training utterances per class respectively. We run in total 20 trials for each experiment in such a way that randomly selected training and testing samples (utterances) do not overlap with each other. Hence in the case of 5 training utterances, the remaining 5 utterances are used for testing and in the case of 7 training utterances, the remaining 3 utterances are used for testing. Results in Figure 1 show that our proposed GT-D outperforms the other tensor decomposition methods for speaker identification.

5. CONCLUSION

We have developed a tensor dictionary learning algorithm for the TUCKER model that incorporates sparsity constraints on the core tensor. To capture the discriminative features, we also learn discriminative dictionaries by applying discriminative constraints on the dictionaries. The in-class and between-class constraints further improve the discriminative power of the dictionaries and thus the classification performance as compared to non-sparse core tensor. The classification results clearly show the ability of our algorithm for maintaining the discerning features of the signals.

6. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[2] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[3] L. R. Tucker, "Some mathematical notes on 3-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Neural Information Processing Systems*, 2001, pp. 556–562.

[5] E. Benetos and C. Kotropoulos, "Non-negative tensor factorization applied to music genre classification," *IEEE Transactions on Audio, Speech and Signal Processing*, vol. 18, no. 8, pp. 1955–1967, November 2010.

[6] A. H. Phan and A. Cichocki, "Extended HALS algorithm for non-negative tucker decomposition and its applications for multiway analysis and classification," *Neurocomputing*, vol. 74, pp. 1956–1969, 2011.

[7] Y-D. Kim and S. Choi, "Non-negative Tucker decomposition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition.*, June 2007, pp. 1 –8.

[8] Y-D. Kim, A. Cichocki, and S. Choi, "Non-negative Tucker decomposition with alpha-divergence," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing.*, 2008, pp. 1829 –1832.

[9] C. F. Caiafa and A. Cichocki, "Block sparse representation of tensors using Kronecker bases," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 2709–2712.

[10] S. Zubair and W. Wang, "Tensor dictionary learning with sparse Tucker decomposition," in *Proc. 18th International Conference on Digital Signal Processing*, July 2013, pp. 1–6.

[11] J. Liu, J. Liu, P. Wonka, and J. Ye, "Spare non-negative tensor factorization using columnwise coordinate descent," *Pattern Recognition*, vol. 45, pp. 649–656, 2012.

[12] Y. Xu, "Alternating proximal gradient method for sparse nonnegative Tucker decomposition," *submitted to SIAM Journal of Scientific Computing*, 2013,[http://arxiv.org/abs/1302.2559].

[13] J. S. Garofolo and et al, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *Linguistic Data Consortium, Philadelphia*, 1995.