SIGNAL CLASSIFICATION BASED ON BLOCK-SPARSE TENSOR REPRESENTATION

Syed Zubair and Wenwu Wang

Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, UK
Emails: s.zubair@surrey.ac.uk, w.wang@surrey.ac.uk

ABSTRACT

Block sparsity was employed recently in vector/matrix based sparse representations to improve their performance in signal classification. It is known that tensor based representation has potential advantages over vector/matrix based representation in retaining the spatial distributions within the data. In this paper, we extend the concept of block sparsity for tensor representation, and develop a new algorithm for obtaining sparse tensor representations with block structure. We show how the proposed algorithm can be used for signal classification. Experiments on face recognition are provided to demonstrate the performance of the proposed algorithm, as compared with several sparse representation based classification algorithms.

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

1. INTRODUCTION

Sparse representation has been used recently in a number of signal processing problems such as source separation [1], computer vision [2], image denoising [3], and signal classification [4]. In classification applications, a class-specific dictionary is typically used to represent the signals from one class with a smaller reconstruction error than the representation for those from other classes. Recently, it was shown that the performance of sparse representation based classification algorithms can be improved by exploiting block sparsity [5], in which the signal is represented sparsely in terms of dictionary blocks rather than the individual atoms. For vectors/matrices data, the block sparsity is utilised by generalizing the non-block sparse representation algorithms such as basis pursuit (BP) [6] and matching pursuit (MP) [7] for block structures. Such extensions have given rise to block matching pursuit (BMP) and block orthogonal matching pursuit (BOMP) [8] algorithms.

The work in [9] shows that, for a dictionary with block structure, representating a test signal in terms of the minimum number of blocks of a dictionary would be a better criterion for classification, as opposed to representing it with the smallest number of atoms. The exploitation of block structure in an underlying dictionary is due to the fact that the training signals of one class often share a common feature that is different from the training examples of another class. Hence a block sparse representation of a test signal is potentially more discriminative as compared to the conventional sparse representation.

In this paper, we extend the concept of block sparsity for high-dimensional tensors and propose a new algorithm for block sparse tensor representation. It has been shown in [10] that tensor based representation has advantages over vector/matrix based representations in retaining the spatial distribution of the data such as the coherence information. Therefore, we propose to introduce the block dictionary structure to tensor representation (based on the well-known TUCKER model) that promotes block sparsity in the coefficients of a tensor. The block structure is useful for building class boundaries in the feature space and helps in improving the classification performance.

We study the performance of this approach within the context of face recognition where the patterns in the images are likely to possess block structure. The remainder of the paper is organised as follows: Section 2 describes classification based on block sparse structure in matrices and Section 3 introduces the concept of block structure in tensors. Section 4 describes our proposed approach. Section 5 discusses the experiments and results while Section 6 concludes the paper.

2. MATRIX BASED BLOCK DICTIONARY AND BLOCK SPARSITY FOR CLASSIFICATION

A class-specific block of a dictionary represents the distribution of variation of the data in that class. For example, in face recognition, the training data of a class may represent not only the facial features but also the facial emotions as well as the lighting conditions. Hence representing the dictionary with block structure has the potential to capture those variations within data. Unlike the classification method in [2] based on conventional sparse representation, the classification with block structure can be achieved by block based sparse representation, where the concept of sparsity in conventional method can be extended naturally to block sparsity.

The work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) of the UK (grant number EP/K014307/1).

In this scenario, an intuitive way to exploit the discriminative distribution of the data in block dictionary in terms of sparsity is the block sparsity. For a dictionary $\mathrm{A} = [\mathrm{A}[1], \ldots, \mathrm{A}[C]]$ with blocks $\mathrm{A}[i]$, where $i = 1, \ldots, C$, the block sparsity is formulated in terms of mixed ℓ_q/ℓ_1 norm as

$$\min \sum_{i=1}^{C} \parallel \mathbf{x}[i] \parallel_q \quad s.t. \ \mathbf{y} = \mathrm{A}\mathbf{x} \tag{1}$$

where $\mathbf{x}[i]$ is the i-th block in the sparse coefficient vector $\mathbf{x}$ corresponding to the dictionary block $\mathrm{A}[i]$. Since each dictionary block corresponds to a specific class, i also represents the class index ranging from 1 to C. This is a convex optimization problem when $q \geq 1$ [5].

Another alternative to exploit sparsity is to minimize the number of reconstructed vectors $\mathrm{A}[i]\mathbf{x}[i]$, as follows [5]:

$$\min \sum_{i=1}^{C} \parallel \mathrm{A}[i]\mathbf{x}[i] \parallel_q \quad s.t. \ \mathbf{y} = \mathrm{A}\mathbf{x} \tag{2}$$

The class label of the test signal can be determined by finding the block of $\mathbf{x}$ that has the maximum ℓ_2 norm. i.e.

$$label(\mathbf{y}) = \arg\max_i \parallel \mathbf{x}[i] \parallel_2 \tag{3}$$

A better alternative for classification is based upon block-sparse reconstruction error [5]. In this case, the label of a test signal $\mathbf{y}$ is determined by

$$label(\mathbf{y}) = \arg\min_i \parallel \mathbf{y} - \mathrm{A}[i]\mathbf{x}[i] \parallel_2 \tag{4}$$

This is the method taken in our work for assigning a class to a test signal.

3. BLOCK DICTIONARY AND SPARSITY FOR TENSORS

This very concept of block dictionary and block sparsity for vectors/matrices can be extended for high dimensional data, e.g. tensors. A N-dimensional tensor $\underline{\mathrm{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ can be represented by the TUCKER model in terms of its dictionaries and the core tensor (coefficients) as:

$$\underline{\mathrm{Y}} = \underline{\mathrm{X}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \cdots \times_N \mathrm{A}^{(N)} \tag{5}$$

where $\mathrm{A}^{(n)} \in \mathbb{R}^{I_n \times M_n}, n = 1, 2, \cdots, N$ are the n-mode dictionary matrices, $\underline{\mathrm{X}} \in \mathbb{R}^{M_1 \times \cdots \times M_N}$ is the core tensor and $\times_n$ is the $n-$mode multiplication between a tensor and a matrix.

The block structure of an n-mode dictionary can be defined as:

$$\mathrm{A}^{(n)} = [\mathrm{A}^{(n)}[1], \ \mathrm{A}^{(n)}[2], \ \cdots, \ \mathrm{A}^{(n)}[C]] \tag{6}$$

where each $\mathrm{A}^{(n)}[i] \in R^{I_n \times M_n^i}$ is the i-th block within $\mathrm{A}^{(n)}$ representing the subspace of class $i = 1, \ldots, C$, M_n^i is the size of the block for class i and n is the mode of the tensor.

Similar to the vector/matrix case, these n-mode block dictionaries can be used to represent the input tensor in terms of the block sparse core tensor. Two approaches can be used to find the block sparse core tensor, i.e. the ℓ_q/ℓ_1 optimization for $q \geq 1$ and greedy algorithms, respectively.

Using optimization techniques, the core tensor with block sparsity can be found by

$$\min \sum_{i=1}^{C} \parallel \underline{\mathrm{X}}[i] \parallel_q \tag{7}$$

Here $\underline{\mathrm{X}}[i] \in \mathbb{R}^{M_1^i \times M_2^i \times \cdots \times M_N^i}$ is the i-th sub-tensor in the core tensor $\underline{\mathrm{X}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$ representing a non-zero core tensor block corresponding to the i-th n-mode block $\mathrm{A}^{(n)}[i]$ of the dictionary $\mathrm{A}^{(n)}$. Figure 1 shows the block sparse structure of a third-order core tensor $\underline{\mathrm{X}}$ in which sub-tensors $\underline{\mathrm{X}}[i]$'s inside the core tensor represent the clusters of non-zeros elements. In case of our n-mode block dictionaries in which each i-th block also represents the i-th class, the solution to (7) gives the minimum numbers of non-zero blocks that lie on the diagonal of the core tensor $\underline{\mathrm{X}}$. These n-mode block dictionaries along with the block sparse core tensor add discriminatory signatures to the core tensor that can be used for classification.

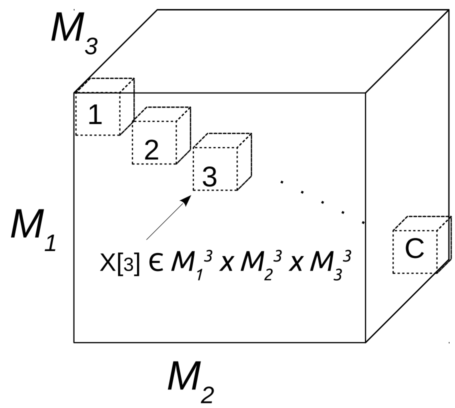

Fig. 1. Block sparse structure of a third-order core tensor $\underline{\mathrm{X}}$.

A test signal $\underline{\mathrm{Y}}^{test}$ can be assigned to a class by first finding its block sparse representation in terms of the n-mode block dictionaries. A class label is then assigned to the tensor signal as:

$$label(\underline{\mathrm{Y}}^{test}) = \arg\min_i \parallel \underline{\mathrm{Y}}^{test} - \underline{\mathrm{X}}[i] \times_1 \mathrm{A}^{(1)}[i] \times_2 \mathrm{A}^{(2)}[i] \cdots \times_N \mathrm{A}^{(N)}[i] \parallel_F \tag{8}$$

where i is the index of the core tensor block and its corresponding dictionary block that give the minimum reconstruction error.

In this paper, we devise a method for calculating block sparse representation for tensors provided that the underlying n-mode dictionaries are given. We use the n-mode block dictionaries to find the subspace of each class. Those block dictionaries are then used to find the block sparse representation of each test signal. Then the signals are classified based on equation (8).

4. METHODOLOGY

Our tensor-based classification method has two main components: (a) n-mode block dictionaries (b) block sparse core tensor.

4.1. Block Dictionaries

We build up an n-mode block dictionary $\mathrm{A}^{(n)}$ comprising of the n-mode blocks of atoms of all the classes representing their respective subspaces. Hence an n-mode block dictionary is represented as in (6). Each individual block $\mathrm{A}^{(n)}[i]$ inside $\mathrm{A}^{(n)}$ is obtained by finding the M_n^i leading left singular vectors of the n-mode unfolded input tensor $\mathrm{Y}^i_{(n)}$ of class i. i.e.

$$\mathrm{A}^{(n)}[i] = SVD(\mathrm{Y}^i_{(n)}, M_n^i) \quad (9)$$

where $i = 1, 2, \ldots, C$ is the block as well as the class index of a signal.

4.2. Block Sparse Core Tensor

The second part of our methodology is to obtain block sparse core tensor where clusters of non-zero values are located in the sub-tensors that constitute the blocks inside the core tensor. Hence we define block sparse representation in terms of the block dictionaries as follows.

Definition 1 (Tensor Block Sparsity):
A multidimensional signal $\underline{\mathrm{Y}} \in R^{I_1 \times I_2 \ldots \times I_N}$ is k-block sparse with respect to the n-mode dictionaries $\mathrm{A}^{(n)} \in \mathbb{R}^{I_n \times M_n}$, if it admits a TUCKER representation as given in equation (5) based only on the blocks of each dictionary $\mathrm{A}^{(n)}[i] \in \mathbb{R}^{I_n \times M_n^i}$ that generates a block sparse core tensor defined as $\underline{\mathrm{X}} = [\ \underline{\mathrm{X}}[1] \ldots \underline{\mathrm{X}}[i] \ldots\ \underline{\mathrm{X}}[k]\ ] \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$ in such a way that

$$\underline{\mathrm{X}}[i] = \underline{\mathrm{X}}\left(M_1^i, M_2^i, \ldots, M_N^i\right) \neq 0 \quad (10)$$

for $k \leq C$, where $\underline{\mathrm{X}}[i] \in \mathbb{R}^{M_1^i \times M_2^i \times \ldots \times M_N^i}$. The sparsity of each block is further defined by $s^i = s_1^i \times s_2^i \times \cdots \times s_N^i$ where s_n^i is the n-mode sparsity of the i-th block. ■

To find the block sparse core tensor, we generalize vectors/matrices based OMP to block tensor OMP (BT-OMP). In this algorithm, the core tensor $\underline{\mathrm{X}}$ is determined with a block sparse structure, similar to the one shown in Figure 1.

The algorithm starts by first finding the block in the n-mode dictionaries which is maximally correlated with the residual $\underline{\mathrm{R}}$ initialized as the input tensor $\underline{\mathrm{Y}}$. This is achieved by locating the block that has the highest correlation with the residual (i.e. the reconstruction error). The index i for the most highly correlated block is found by:

$$i^t = \arg\max_i \ \| \underline{\mathrm{R}} \times_1 \mathrm{A}^{(1)^T}[i] \times_2 \cdots \times_N \mathrm{A}^{(N)^T}[i] \| \quad (11)$$

where i^t is the index of the block at iteration t containing an atom which has maximum correlation with the input signal.

From the chosen block of the dictionary, the atoms which are maximally correlated with the residual tensor are chosen by:

$$[m_1^t, \ldots, m_N^t] = \arg \max_{m_1, \ldots, m_N} \ \| \underline{\mathrm{R}} \times_1 \mathrm{A}^{(1)^T}[i^t](:, m_1) \times_2 \cdots \times_N \mathrm{A}^{(N)^T}[i^t](:, m_N) \| \quad (12)$$

where m_n is the index of the atom in the i-th block of the n-mode dictionary. This helps to select those atoms which generate the sub-tensors while calculating the core tensor.

After t iterations, the selected indices of blocks and the atoms therein are combined to make the n-mode dictionary blocks $\mathrm{B}^{(n)}[i^t] \in \mathbb{R}^{I_n \times \mathcal{M}_n^i}$, as follows:

$$\mathrm{B}^{(n)}[i^t] = \mathrm{A}^{(n)}[i^t](:, \mathcal{M}_n^i) \quad (13)$$

where $\mathcal{M}_n^i = \mathcal{M}_n^i \cup [m_n^t]$ is the set of indices of atoms belonging to the i-th block at iteration t.

These n-mode dictionary blocks with a reduced number of atoms are then used to find the core tensor. In vector form, the formulation of the problem for the calculation of the core tensor $\underline{\mathrm{E}}$ can be represented as:

$$\mathbf{e} = \arg\min_{\mathbf{u}} \ ||(\mathrm{B}^{(N)}[i^t] \otimes \cdots \otimes \mathrm{B}^{(1)}[i^t])\mathbf{u} - \mathbf{y}||_2^2 \quad (14)$$

where $\mathbf{y} \in \mathbb{R}^I (I = \prod_{n=1}^N I_n)$ is the vectorized form of the input tensor $\underline{\mathrm{Y}}$, , $\mathbf{e} \in \mathbb{R}^m (m = \prod_{n=1}^N \mathcal{M}_n^i)$ is the vectorized form of $\underline{\mathrm{E}}$ and $\mathrm{B}^{(n)}[i^t]$ is defined in equation (13).

Once $\mathbf{e}$ is found according to (14), its tensorised form $\underline{\mathrm{E}}$ is then used to update the residual, as follows:

$$\underline{\mathrm{R}} = \underline{\mathrm{Y}} - \underline{\mathrm{E}}[i^t] \times_1 \mathrm{B}^{(1)}[i^t] \times_2 \cdots \times_N \mathrm{B}^{(N)}[i^t] \quad (15)$$

For the given n-mode dictionaries, the core tensor is calculated with block sparsity until the stopping criterion is reached.

The problem of (14) can be addressed in two different ways, namely, the least squares method and an iterative method. The least squares solution is derived based on the vector form, where the signal in terms of dictionary blocks can be approximated by:

$$\hat{\mathbf{y}} = (\mathrm{B}^{(N)}[i^t] \otimes \cdots \otimes \mathrm{B}^{(1)}[i^t])\mathbf{e}, \quad (16)$$

Denote $\mathrm{B} = \mathrm{B}^{(N)}[i^t] \otimes \cdots \otimes \mathrm{B}^{(1)}[i^t]$, then (16) can be written as:

$$\hat{\mathbf{y}} = \mathrm{B}\ \mathbf{e} \quad (17)$$

therefore, we have

$$\mathbf{e} = [\mathrm{B}^T\mathrm{B}]^{-1}\mathrm{B}\mathbf{y} \tag{18}$$

Alternatively, we can use an iterative method to find the core tensor, as follows. We notice from (18) that $[\mathrm{B}^T\mathrm{B}]\mathbf{e} = \mathrm{B}\mathbf{y}$, which can be represented in tensor form as

$$\underline{\mathrm{E}} \times_1 \mathrm{B}^{(1)T}\mathrm{B}^{(1)} \times_2 \cdots \times_N \mathrm{B}^{(N)T}\mathrm{B}^{(N)} = \underline{\mathrm{Y}} \times_1 \mathrm{B}^{(1)} \times_2 \cdots \times_N \mathrm{B}^{(N)} \tag{19}$$

In unfolded matrix form, (19) is written as

$$\mathrm{B}^{(1)T}\mathrm{B}^{(1)}\mathrm{E}_{(1)}(\mathrm{B}^{(N)T}\mathrm{B}^{(N)} \otimes \cdots \otimes \mathrm{B}^{(2)T}\mathrm{B}^{(2)}) = \mathrm{B}^{(1)}\mathrm{Y}_{(1)}(\mathrm{B}^{(N)} \otimes \cdots \otimes \mathrm{B}^{(2)})^T \tag{20}$$

Let us define $\underline{\mathrm{P}}^{(1)} = \underline{\mathrm{E}} \times_1 \mathrm{I} \times_2 \cdots \times_N \mathrm{B}^{(N)T}\mathrm{B}^{(N)}$ and $\underline{\mathrm{Q}} = \underline{\mathrm{Y}} \times_1 \mathrm{B}^{(1)} \times_2 \cdots \times_N \mathrm{B}^{(N)}$, then we can write (20) as:

$$\mathrm{B}^{(1)T}\mathrm{B}^{(1)}\mathrm{P}^{(1)}_{(1)} = \mathrm{Q}_{(1)} \tag{21}$$

where $\mathrm{P}^{(1)}_{(1)}$ and $\mathrm{Q}_{(1)}$ are the mode-1 unfolded forms of $\underline{\mathrm{P}}^{(1)}$ and $\underline{\mathrm{Q}}$, respectively. This can be solved by

$$\mathrm{P}^{(1)}_{(1)} = [\mathrm{B}^{(1)T}\mathrm{B}^{(1)}]^{\dagger}\ \mathrm{Q}_{(1)} \tag{22}$$

where $\dagger$ is the pseudo-inverse. Now the solution of $\mathrm{P}^{(1)}_{(1)}$ can be used to write its mode-2 unfolded form as:

$$\mathrm{P}^{(1)}_{(2)} = \mathrm{B}^{(2)T}\mathrm{B}^{(2)}\mathrm{E}_{(2)}(\mathrm{B}^{(N)T}\mathrm{B}^{(N)} \otimes \cdots \otimes \mathrm{B}^{(3)T}\mathrm{B}^{(3)} \otimes \mathrm{I}) \tag{23}$$

and

$$\mathrm{P}^{(1)}_{(2)} = \mathrm{B}^{(2)T}\mathrm{B}^{(2)}\ \mathrm{P}^{(2)}_{(2)} \tag{24}$$

where $\underline{\mathrm{P}}^{(2)} = \underline{\mathrm{E}} \times_1 \mathrm{I} \times_2 \mathrm{I} \times \cdots \times_N \mathrm{B}^{(N)T}\mathrm{B}^{(N)}$ and again $\mathrm{P}^{(2)}_{(2)}$ is the mode-2 unfolded form of $\underline{\mathrm{P}}^{(2)}$ which can be solved by multiplying the pseudo-inverse of $\mathrm{B}^{(2)T}\mathrm{B}^{(2)}$ with $\mathrm{P}^{(1)}_{(2)}$ as done for the solution of $\mathrm{P}^{(1)}_{(1)}$ in equation (22). In this way, by repeating this procedure N times, we finally find $\underline{\mathrm{E}}$.

The overall algorithm for BT-OMP is given in Algorithm 1.

Algorithm 1: BT-OMP

Task: Find block sparse representation $\underline{\mathrm{Y}} = \underline{\mathrm{X}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \cdots \times_N \mathrm{A}^{(N)}$ with $\underline{\mathrm{X}} = [\ \underline{\mathrm{X}}[1]\ \ \underline{\mathrm{X}}[2]\ \ldots\ \underline{\mathrm{X}}[C]\ ]$ and $\underline{\mathrm{X}}[i] = \underline{\mathrm{X}}\left(M_1^i, M_2^i, \ldots, M_N^i\right)$ such that $x_{m_1 \ldots m_N} = 0\ \forall\ (m_1, \ldots, m_N) \notin M_1^i \times \ldots \times M_N^i,\quad i = 1, \ldots, C.$
$\underline{\mathrm{X}} = \underline{\mathrm{E}}$.
Require: n-mode dictionaries $\mathrm{A}^{(n)} \in R^{I_n \times M_n}$, for $n = 1, \ldots, N$, having dictionary blocks $\mathrm{A}^{(n)}[i] \in R^{I_n \times M_n^i}$, input signal $\underline{\mathrm{Y}}$, the maximum number of non-zero coefficients $t_{max} \leq s^i$ in a block $\underline{\mathrm{X}}[i]$, and the tolerance ϵ.
Output: $\underline{\mathrm{X}} = \underline{\mathrm{E}}$ and $\{b_i \mid i \in \{1, \ldots, C\}\}$ where b_i is the index of the i-th element in set b.
Initialization:
$b = [\emptyset],\ \ \mathcal{M}_n^i = [\emptyset],\ \ \underline{\mathrm{R}} = \underline{\mathrm{Y}},\ \ \underline{\mathrm{X}} = \underline{0},\ \ t = 1;$
while ($|b| < C$ and $||\underline{\mathrm{R}}||_F > \epsilon$) **do**

1. Find block index i^t from (11).
2. Select indices of atoms in the n-mode dictionary blocks i^t using (12).
3. $b = b \cup i^t,\ \ \mathcal{M}_n^i = \mathcal{M}_n^i \cup [m_n^t],\ \ \mathrm{B}^{(n)}[i^t] = \mathrm{A}^{(n)}[i^t](:, \mathcal{M}_n^i)$, for all $n = 1, \ldots, N$;
4. $\mathbf{e} = \arg\min_{\mathbf{u}}\ ||(\mathrm{B}^{(N)}[i^t] \otimes \cdots \otimes \mathrm{B}^{(1)}[i^t])\mathbf{u} - \mathbf{y}||_2^2$, and tensorise $\mathbf{e}$ to $\underline{\mathrm{E}}[i^t]$;
5. $\underline{\mathrm{R}} = \underline{\mathrm{Y}} - \underline{\mathrm{E}}[i^t] \times_1 \mathrm{B}^{(1)}[i^t] \times_2 \cdots \times_N \mathrm{B}^{(N)}[i^t]$;
6. $t = t + 1$;

end while
return b, $\underline{\mathrm{E}}$;

5. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed algorithm for face recognition, and compare it with other state-of-the-art sparse dictionary learning based classification methods, such as Block Sparse Representation Classification [5] (B-SRC), Sparse Representation Classification (SRC) [2] and the Nearest Subspace (NS) [11], which are all vector/matrix based methods.

In B-SRC, each test signal is represented in terms of sparse blocks as given in (1) and (2) respectively, with an underlying block dictionary. We denote the B-SRC classification result based upon (1) by P_{ℓ_2/ℓ_1} and the one on (2) by P'_{ℓ_1/ℓ_1}. A label is assigned to the test signal based upon (4). For $q = 1$, equation (1) represents the SRC method where the signal representation as well as the classification is based upon sparse representation in terms of atoms, hence it is denoted by P_{ℓ_1/ℓ_1}. In the NS method, the classification method looks for a subject whose underlying subspace is closest to the given test signal.

In the evaluations, we use the Extended Yale B database [12], which consists of 2414 frontal face images of $C = 38$ subjects. For each subject, there are approximately 64 images of size 192×168. These images are captured under various laboratory controlled lighting conditions.

For this database, we first find the n-mode block dictionaries wherein each block represents the subspace of each class. For $C = 38$ subjects, we have 38 blocks of dictionary. For this purpose, the database images are first down-sampled from 192×168 to 96×84. The down-sampled images are then stacked together to make a 3-D input tensor of size $96 \times 84 \times I_3$, where $I_3 \in [9,\ 18,\ 25,\ 32]$, depending upon a varying number of training signals to train the dictionary. The size of each dictionary block is $I_n \times M_n^i = I_n \times 32$, i.e., for each mode, we select only 32 left leading eigenvectors of the n-mode unfolded input tensor. The core tensor to be calculated is of size $M_n = I_n$ with $M_n^i = 32$ and the fixed mode block sparsity, $\mu[i] = s_n[i]/M_n^i$ is set to 0.62.

With these settings, we run our proposed algorithm for a different number of training signals for each class, i.e. [9, 18, 25, 32] in such a way that the training and testing data of each class do not overlap. We also run the experiment for other algorithms with the same amount of training data. For all other vector/matrix based algorithms, the down-sampled images are first converted to vectors. Those image-vectors are projected to the first 132 principal components of the training data covariance matrix. The choice of 132 is made by following the strategy of the base-line method [5] that gives the best classification result. For chosen training samples with rest of the dataset signals used as testing samples, we run 10 trials of our experiment for all the methods and the classification results are shown in Figure 2.

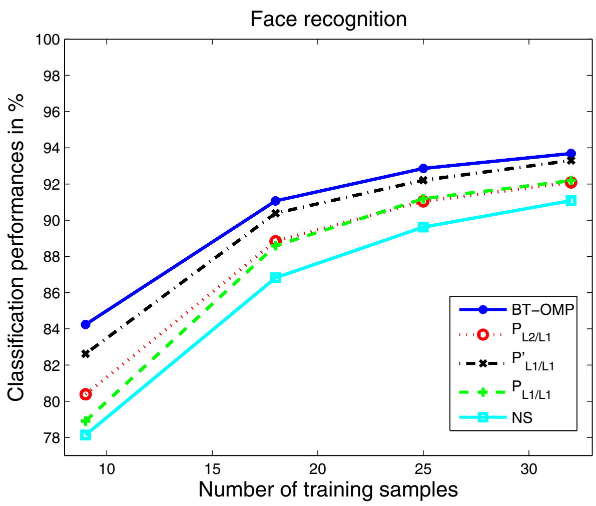

Fig. 2. Classification rates of different algorithms for different number of training examples.

This figure shows that the tensor based decomposition along with block dictionary and block sparse core tensor (i.e. our proposed BT-OMP algorithm) has superior classification performance over its vector/matrix based counter-part as well as other state-of-the-art algorithms.

6. CONCLUSION

In this paper, we have introduced the concept of block sparsity for tensors with an underlying block dictionary. The experiments show that this notion introduces a discriminative feature to the signal structure that helps to improve the classification performances. The results show the supremacy of tensor based structures over vector/matrix based methods for classification applications.

7. REFERENCES

[1] Y. Li, S. Amari, A. Cichocki, D. W. C. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 2, pp. 423–437, 2006.

[2] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[3] 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.

[4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Discriminative learned dictionaries for local image analysis," in *IEEE conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[5] E. Elhamifar and R. Vidal, "Robust classification using structured sparse representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2011, pp. 1873–1879.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[7] M. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. on Signal processing*, vol. 41, pp. 3397–3415, 1993.

[8] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, June 2010.

[9] E. Elhamifar and R. Vidal, "Block-sparse recovery via convex optimization," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4094–4107, Aug. 2012.

[10] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3D non-negative tensor factorization," in *IEEE International Conference on Computer Vision*, October 2005, vol. 1, pp. 50 – 57.

[11] J. Ho, M. H. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003, vol. 1, pp. 11–18.

[12] K. C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under varaible lighting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, May 2005.