FAST COMPRESSIVE SENSING OF HIGH-DIMENSIONAL SIGNALS WITH TREE-STRUCTURE SPARSITY PATTERN

Chun-Shien Lu[1] *and Wei-Jie Liang*[1,2]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2]Department of Mathematics, National Cheng-Kung University, Tainan, Taiwan

ABSTRACT

Compressive sensing of multi-dimensional signals (tensors) only receives limited attention. Separable sensing and proper sparsity pattern play two key roles for compressive sensing of tensors to be feasible and efficient. In view of inherent characteristic of 2D images and 3D videos, we propose the use of tree-structure sparsity pattern in tensor compressive sensing and develop a multiway tree-structure sparsity pattern OMP algorithm in this paper. Experimental results demonstrate the effectiveness of our method in terms of recovery quality and speed.

Index Terms— Compressed sensing, Kronecker structure, Matching pursuit, Sparsity, Tensor, Tucker model

1. INTRODUCTION

1.1. Background

Compressive sensing (CS) [1, 2, 3] of sparse signals in achieving simultaneous data acquisition and compression has been extensively studied in the past few years. In the context of CS, we usually let x denote a K-sparse signal of length M to be sensed, let ϕ of dimensionality $I \times M$ represent a sampling matrix, and let y be the measurement of length I, where $K < I < M$. At the encoder, random projection, defined as:

$$y = \phi x, \tag{1}$$

is conducted on x via ϕ to obtain the measurement vector y. The measurement rate in CS is defined as $0 < \frac{I}{M} < 1$ At the decoder, the original signal x to be sensed can be recovered to a certain extent by means of convex optimization or greedy algorithms.

Compressive sensing has been widely studied for 1D signals via random projection in Eq. (1). However, compressive sensing of higher dimensional signals beyond 2D has only received little attention.

1.2. Motivation and Our Contributions

In this paper, we study a new compressive sensing scheme for high-dimensional signals (or tensors and multiway arrays) by exploiting the inherent tree-structure sparsity pattern based on Tucker model and the equivalent Kronecker structure. Sensing and sparse recovery affect the performance (reconstruction quality and speed) of a compressive sensing algorithm. The ways of sensing include joint sensing [4], independent sensing, and independent but separable sensing (with different signal decompositions). Both joint sensing and independent sensing incur tremendous storage and computation overhead. In this paper, separable sensing, as presented in [5, 6] and adopted in [7], is employed.

For sparse recovery of multiway arrays, sparsity pattern plays an important role. With an eye on the natural characteristic of tree-structure relationship among wavelet coefficients that are popularly used to represent 2D images and 3D videos, we propose to explore tree-structure sparsity pattern (TSSP) in tensor compressive sensing. Support detection via TSSP possesses the advantage of reducing the overhead of calculating the correlation between the current residual and each possible candidate support that is a necessary step in matching pursuit. Nevertheless, tree-structure sparsity pattern and Kronecker structure (block-based) are not consistent. If Kronecker dictionary cannot be adopted during the sparse recovery process, recovery speed will be an obstacle. To address this issue, we propose to modify the fast least-square solution in [7] to fit TSSP for our use. Extensive results demonstrate that our method is able to capture more significant supports while achieving faster recovery than [7].

2. RELATED WORK

In [4], Durate and Baraniuk introduce Kronecker product to model multidimensional compressive sensing of signals. The authors prove mutual incoherence and restricted isometric property for the resultant Kronecker sensing matrix and Kronecker dictionary. Although the presented Kronecker compressive sensing (KCS) method seems to be reasonable, its practical use is greatly prohibited. This is due to the vectorization of multidimensional signals and the use of joint sensing involving a (very) large Kronecker product-based sensing matrix, leading to unacceptable overhead of storage and computation.

In [8], a multiway compressive sensing (MWCS) method

978-1-4799-5403-2/14/$31.00 ©2014 IEEE

for sparse and low-rank tensors is proposed. Although MWCS achieves more efficient reconstruction, its performance relies heavily on tensor rank estimation, which is NP-hard. Another compressive sensing method for higher-order tensors, called generalized tensor compressive sensing (GTCS), has been recently proposed in [9]. GTCS is demonstrated to be comparable to KCS in recovery accuracy and be greatly faster than KCS in recovery speed.

While the previous studies consider some tensor operations like Kronecker product, CP decomposition, and Tucker model within the framework of compressive sensing, the sparsity pattern inherent in the tensor (like 2D image and 3D video) has not been fully explored. This may explain their limited recovery performance. Recently, Caiafa and Cichocki [7] exploit Kronecker product and block sparsity to develop a so-called N-BOMP (N-way block OMP) algorithm in realizing tensor recovery from few sensing measurements by block sparsity-based OMP [10]. The authors also prove the equivalence between Tucker model and Kronecker representation for multiway arrays; thus, Kronecker structure can be used to solve the Tucker model-based underdetermined linear systems within compressive sensing. However, we find, as also indicated in subsection 7.2.1 of [7], that for an 2D image it is pre-processed in advance to possess perfect block sparsity pattern. Here perfect sparsity pattern means that the important/significant coefficients in some transform domain fall within the specified block sparsity pattern while other insignificant coefficients are entirely removed. Under the situation, N-BOMP is able to obtain reconstruction quality far better than the existing tensor CS algorithms. The key weakness of [7] will be further discussed in Sec. 3.3.

3. COMPRESSIVE SENSING FOR TENSORS

In this section, we describe the proposed method for tensors via OMP. We investigate a tree-structure sparsity pattern in tensors to capture the inherent characteristics in tensors (*e.g.,* images and videos) for support detection. We also investigate how to incorporate the tree-structure sparsity pattern with the fast least-square solution in N-BOMP, which is originally designed to specially accommodate block-sparsity pattern.

3.1. System Model

We consider how to generalize 1D compressive sensing in Eq. (1) to multiway arrays. Let the 3-dimensional tensor (video sequence) be denoted as $\mathbf{X} \in \Re^{M_1 \times M_2 \times M_3}$. Let the Kronecker sensing matrix be denoted as $\mathbf{\Phi} = \Phi_3 \otimes \Phi_2 \otimes \Phi_1$, which can be rewritten as $\mathbf{\Phi} = I_d \otimes \Phi_2 \otimes \Phi_1$ if we separably sensing each video frame using $\Phi_2 \otimes \Phi_1$ via separable sensing, where I_d denotes an identity matrix, $\Phi_1 \in \Re^{I_1 \times M_1}$, $\Phi_2 \in \Re^{I_2 \times M_2}$, and $I_1 < M_1$ and $I_2 < M_2$. Then, compressive sensing of a tensor $\mathbf{X}$ via $\mathbf{\Phi}$ can be expressed via Tucker representation as:

$$\mathbf{Y} = \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \times_3 I_d, \tag{2}$$

where $\mathbf{Y} \in \Re^{I_1 \times I_2 \times I_3}$ denotes the measurement. Please note that $\mathbf{X}$ in Eq. (2) is called core tensor in the context of Tucker model and its size is smaller than $\mathbf{Y}$. This is, however, not the case in compressive sensing.

The tensor $\mathbf{X}$ has to satisfy a certain sparsity such that it is possible to recover $\mathbf{X}$ from (far) few measurements $\mathbf{Y}$ in the underdetermined system, indicated in Eq. (2), where $|\mathbf{Y}| < |\mathbf{X}|$ and $|\cdot|$ denotes the cardinality of a set. Furthermore, let the 3-dimensional tensor $\mathbf{X}$ be sparse with respect to Ψ_1, Ψ_2, and Ψ_3. The sparse representation of $\mathbf{X}$ can be represented via Kronecker dictionary as:

$$vec(\mathbf{X}) = (\Psi_3 \otimes \Psi_2 \otimes \Psi_1)\boldsymbol{\theta}, \tag{3}$$

where $\Psi_3 \otimes \Psi_2 \otimes \Psi_1$ denotes a Kronecker dictionary, and the sparsity is K. A 1D vector, vec($\mathbf{X}$), denoting the vectorization of $\mathbf{X}$, will be expressed as $\mathbf{x}$ hereafter, and $\boldsymbol{\theta}$ is the sparse representation of $\mathbf{x}$ with respect to the Kronecker dictionary.

According to [7], Eq. (2) can be derived as:

$$\mathbf{y} = (I_d \otimes \Phi_2 \otimes \Phi_1)\mathbf{x}. \tag{4}$$

Plugging Eq. (3) into Eq. (4), we have

$$\begin{aligned} \mathbf{y} &= (I_d \otimes \Phi_2 \otimes \Phi_1)(\Psi_3 \otimes \Psi_2 \otimes \Psi_1)\boldsymbol{\theta} \\ &= (\Psi_3 \otimes \Phi_2\Psi_2 \otimes \Phi_1\Psi_1)\boldsymbol{\theta}. \end{aligned} \tag{5}$$

Let $D_1 = \Phi_1\Psi_1$, $D_2 = \Phi_2\Psi_2$ and $D_3 = \Psi_3$, then Eq. (5) can be rewritten via Tucker model as:

$$\mathbf{Y} = \mathbf{\Theta} \times_1 D_1 \times_2 D_2 \times_3 D_3, \tag{6}$$

where θ is the vectorization of $\mathbf{\Theta}$.

3.2. Sparse Signal Recovery for Tensors

Orthogonal matching pursuit (OMP) has received much attention for sparse signal recovery due to its efficiency. We also consider OMP for tensors here. OMP is basically composed of two steps: support detection and least-square (LS) solution. For support detection, the maximum correlation, defined as:

$$\arg\max_{[i_1,i_2,i_3]} |\mathbf{Y} \times_1 D_1(:,i_1)^T \times_2 D_2(:,i_2)^T \times_3 D_3(:,i_3)^T|, \tag{7}$$

is exploited to collect the supports to form the support set $\mathcal{I}_n$, which is a set of indices for mode n ($n = 1, 2, 3$). Then, the values of supports are obtained by solving the LS problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|(B_3 \otimes B_2 \otimes B_1)\boldsymbol{\theta} - \mathbf{y}\|, \tag{8}$$

where B_n denote the submatrices obtained by restricting the mode-n dictionaries D_n to columns, indicated by $\mathcal{I}_n$ (*i.e.,* $B_n = D_n(:, \mathcal{I}_n)$). The solution of Eq. (8) is then given by $(B^T B)^{-1} B^T \mathbf{y}$, where $B = B_3 \otimes B_2 \otimes B_1$.

3.3. Tree-Structure Sparsity Pattern in Tensors

The sparsity pattern in $\mathbf{\Theta}$ will determine recovery performance and speed. It may not be so important if $\mathbf{X}$ is 1D, though block sparsity is verified to be better [10]. Nevertheless, sparsity patterns for 2D images and 3D video sequences have to capture the inherent characteristics in order to attain better reconstruction quality. In [7], the authors argue that non-zero entries tend to gather together for tensors and suggest adopting block sparsity [10]. In fact, we find from [7] that we would rather say that block sparsity is adopted in order to entirely utilize Kronecker dictionary for fast recovery. Block sparsity, however, is not the best choice to describe the inherent structure of images and videos. On the contrary, tree-structure sparsity pattern is more suitable to describe images or videos, as evidenced in conventional image/video compression standards. We show that the tree-structure sparsity pattern is better than the pure block sparsity pattern in reconstructing images/videos within the framework of tensor compressive sensing. The advantage is confirmed by capturing more significant supports (*i.e.*, larger non-zero entries in the wavelet domain).

We adopt the wavelet, as provided in the source code of [7], as the dictionary. In our method, the tree-structure sparsity pattern (TSSP) is defined to link the significant wavelet coefficients across scales. Therefore, the linked significant wavelet coefficients at the neighboring scales define the parent-child relationship.

Let's take 2D image as an example to describe how to build a TSSP. Suppose S-scale wavelet decompositions are conducted on the image to obtain a series of subbands as:

$$\{LL_S, LH_S, HL_S, HH_S, LH_{S-1}, HL_{S-1}, HH_{S-1}, ...HH_1\}.$$

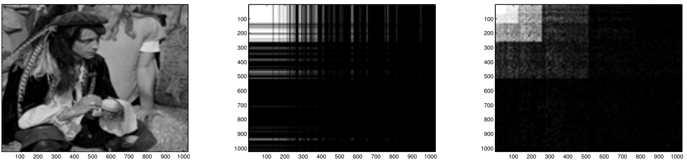
First, since LL_S contains the most important information for signal recovery, the entries in LL_S are included in the set of supports unconditionally.

Second, the siblings of LL_S, including LH_S, HL_S, and HH_S, will be considered as usual support detection in compressive sensing to create the roots of multiple trees. More specifically, for each entry in the siblings of LL_S, we calculate the correlation defined in Eq. (7) without needing to consider the maximum operation in our case. If the result is larger than a threshold T_S, then it is said that this entry is a support and its children are put in the set Q for support detection later. We repeat the above procedure until all entries belonging to the sibling of LL_S are checked.

Third, we take an entry from the set Q and calculate the correlation. Again, if the correlation is larger than a threshold $T_S/2^{S-s}$, where $1 \leq s < S$, then it is said that an element of a tree is found and its children are put in the set Q for support detection later. We repeat the above procedure until the Q is empty to finally generate the tree-structure sparsity pattern.

In our method, the thresholds $T_S/2^{S-s}$'s are defined to depend on the scaling factor and parent-child relationship is conventionally defined for dyadic wavelet decomposition.

Fig. 1 shows an example of detected block-sparsity pattern in N-BOMP and TSSP in our method.



(a) (b) (c)

Fig. 1. (a) Man. (b) Block-sparsity pattern. (c) TSSP.

3.3.1. Remark

In N-BOMP, Eq. (7) should be calculated at each iteration for one support detection because there is no hint based on block-sparsity pattern to find the next support. However, for the tree-structure sparsity pattern presented here, the locations of supports possibly follow the path of a growing tree; thus, our method does not need to spend time to calculate the maximum correlation to find a support individually. As a result, our method is faster than N-BOMP in support detection. This is particularly obvious for images of large sizes.

3.4. Incorporate TSSP and LLT

The solution of Eq. (8) is given by $\hat{\boldsymbol{\theta}} = [B^TB]^{-1}B^T\mathbf{y}$, which implies that $[B^TB]\hat{\boldsymbol{\theta}} = B^T\mathbf{y}$, where $B = B_2 \otimes B_1$. This is a large scale matrix and needs high computation cost to calculate pseudo-inverse matrix since the size of B^TB is $|\mathcal{I}_1||\mathcal{I}_2| \times |\mathcal{I}_1||\mathcal{I}_2|$ for 2D cases. Hence, we utilize the Cholesky factorization from N-BOMP [7] to deal with this problem. The main idea is based on unfolding to divide the whole problem into smaller problems. More specifically, we unfold $[B^TB]\hat{\boldsymbol{\theta}} = B^T\mathbf{y}$ to be $B_1^TB_1\hat{\Theta}_{(1)}B_2^TB_2 = B_1^TY_{(1)}B_2$, where $\hat{\Theta}_{(1)}$ and $Y_{(1)}$ are mode-1 of $\hat{\mathbf{\Theta}}$ and $\mathbf{Y}$, respectively. Let $Z_{(1)} = \hat{\Theta}_{(1)}B_2^TB_2$ and $P_{(1)} = B_1^TY_{(1)}B_2$ be mode-1 matrix of $\mathbf{Z}$ and $\mathbf{P}$, respectively. We have:

$$B_1^TB_1Z_{(1)} = P_{(1)}. \tag{9}$$

Note that the size ($|\mathcal{I}_1| \times |\mathcal{I}_1|$) of $B_1^TB_1$ becomes smaller than that of the original one $B_1^TB_1$. Now, we can utilize Cholesky factorization of $B_1^TB_1$ to more easily solve Eq. (9) to obtain the solution $\mathbf{Z} = \hat{\mathbf{\Theta}} \times_1 I_d \times_2 B_2^TB_2$. Based on the similar unfolding strategy, we unfold mode-2 matrix of $\mathbf{Z}$ as $B_2^TB_2\hat{\Theta}_{(2)} = Z_{(2)}$, where the size of $B_2^TB_2$ is $|\mathcal{I}_2| \times |\mathcal{I}_2|$. For simplicity, we call the above process, LLT.

It is, however, not intuitive to incorporate TSSP and LLT. As indicated in Eq. (9), the rank of $B_1^TB_1$ is $|\mathcal{I}_1|$, where $B_1 \in \Re^{I_1 \times |\mathcal{I}_1|}$. Hence, the condition $|\mathcal{I}_1| < I_1$ must be satisfied so as to make $B_1^TB_1$ invertible. Similarly, to ensure that $B_2^TB_2$ is invertible, $|\mathcal{I}_2| < I_2$ is also a must.

Table 1. Comparison between our method and N-BOMP under target sparsity K and measurement rates of $20\% \sim 40\%$.

$K = I/2$	Ours			N-BOMP		
	20%	30%	40%	20%	30%	40%
No. of detected supports	104881	157360	209951	105093	157652	210385
Hit Ratio	0.4772	0.4831	0.4855	0.4529	0.4735	0.4901
PSNR (dB)	23.0119	23.9915	25.0218	22.2796	23.8300	25.1260
SSIM	0.4939	0.5411	0.5854	0.4526	0.5214	0.5798
Execution Time (seconds)	14.0187	21.9679	22.5372	22.1308	34.9959	49.1732

$K = I/5$	Ours			N-BOMP		
	20%	30%	40%	20%	30%	40%
No. of detected supports	41953	62944	83981	42106	63181	84216
Hit Ratio	0.4709	0.4783	0.4822	0.4526	0.4687	0.4819
PSNR (dB)	23.6949	24.2718	25.0359	23.6225	24.3654	25.2298
SSIM	0.5369	0.5664	0.5891	0.5211	0.5652	0.6008
Execution Time (seconds)	8.8047	9.9713	14.6362	14.4252	22.1966	30.1810

However, when we collect the indices of supports from TSSP, $|\mathcal{I}_1|$ (or $|\mathcal{I}_2|$) may exceed I_1 (or I_2). To avoid such situation, we divide the whole problem into a few smaller problems. More specifically, the indices of supports are collected and controlled to satisfy the above constraint, and are solved to obtain partial solutions. Finally, these partial solutions are combined to form the final complete solution.

4. SIMULATION RESULTS

Since our method is an improvement over N-BOMP [7] and the source code of N-BOMP is available (http://web.fi.uba.ar/∼ccaiafa/Cesar/N-BOMPdemop.html), comparison between both is the focus here. All simulations were conducted on PC equipped with Windows 7 with 3.4 GHz Intel Core i7 CPU and 8GB RAM. In the following, we shall take 2D image as an example, *i.e.*, compressive imaging (CI) in [7], for our simulations due to space limit.

4.1. Parameter Setting

Parameter setting is similar to [7] in that the sensing matrix was random Gaussian, the Daubechies separable wavelet was used as the sparsifying basis, and OMP was adopted for CS recovery. However, unlike [7] in dealing with CI, the image to be sensed is not pre-processed in advance[1].

4.2. Performance Evaluation and Comparison

Performance evaluation and comparison were conducted in terms of recovery quality (in PSNR and SSIM [11]), execution time, and hit ratio (defined as the number of true supports per number of detected supports). Some standard images, including, Lena, Baboon, F16, Peppers, Goldenhill, sailboat, and Man were used as the targets to be sensed and recovered. Except the Man image of size 1024×1024, the other images are with size of 512×512. Thus, $M_1 = M_2 = 1024$ or 512. The measurement rates were set to 20%, 30% and 40%, respectively. The target sparsity is defined to be $K = \frac{I}{r}$, where $I = I_1 \times I_2$ and $2 \le r \le 5$. Table 1 shows the results for Fig. 1(a). Each entry was obtained from averaging 60 results. We observe similar results for other images.

We can find that the execution time of our method is less than that of N-BOMP. The time saving is particularly obvious for large images. This is because our method only needs to calculate the maximum correlation at each layer but N-BOMP requires such calculation for each support detection.

As for reconstruction quality, we find that our method is comparable to N-BOMP. The reasons may be (1) our method extracts the number of supports smaller than that of N-BOMP, and (2) as mentioned previously, we only do maximum correlation at each layer such that the resultant residual may affect support detection and recovery.

The hit ration is presented as the criterion to verify the correctness of support detection. We can observe from Table 1 that our tree-structure sparsity pattern is able to detect more significant supports (with large wavelet coefficients) than N-BOMP does. This is particularly obvious for complex images.

Finally, it is worth noting that the recovery quality of our method is improved faster than that of N-BOMP with the increase of detected supports under the same measurement rate. But the overall recovery performance will be indistinguishable for both methods when the number of detected supports are large enough, as indicated in Table 1.

5. CONCLUSIONS

In view of inherent characteristic of 2D images and 3D videos, we propose to use tree-structure sparsity pattern in compressive sensing of tensors and develop a multiway OMP algorithm based on TSSP. Our future work would be to develop our own sparse signal recovery algorithm for tree-structure sparsity pattern since the incorporation of TSSP and LLT still does not exhibit sufficiently good recovery.

[1] For example, in subsection 7.2.1 of [7], the authors propose to keep the largest $25,000$ wavelet coefficients for a 2D image of size 1024×1024 by thresholding. Such way can perfectly guarantee to sample/capture significant coefficients in a block manner, as done in N-BOMP, but this is impractical.

6. REFERENCES

[1] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[2] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[3] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] M. Duarte and R. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. on Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[5] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 449–452, 2009.

[6] Y. Rivenson and A. Stern, "Practical compressive sensing of large images," in *16th Int'l Conf. on Digital Signal Processing*, 2009.

[7] C. F. Caiafa and A. Cichocki, "Computing sparse representations of multidimensional signals using kronecker bases," *Neural Computation Journal*, vol. 25, no. 1, pp. 186–220, 2013.

[8] N. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 757–760, 2012.

[9] Q. Li, D. Schonfeld, and S. Friedland, "Generalized tensor compressive sensing," in *IEEE ICME*, 2013, pp. 1–6.

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

[11] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.