A CLUSTERING BASED FRAMEWORK FOR DICTIONARY BLOCK STRUCTURE IDENTIFICATION

Ender M. Eksioglu

Electronics and Communications Engineering Department
Istanbul Technical University
Istanbul, Turkey

ABSTRACT

Sparse representations over redundant dictionaries offer an efficient paradigm for signal representation. Recently block-sparsity has been put forward as a prior condition for some sparse representation applications, where the coefficients of the sparse representation occur in blocks rather than being distributed randomly over the sparse vector. Block-sparse representation algorithms, which are extensions of the regular sparse representation algorithms have been developed. However, these algorithms work under the assumption that both the dictionary and its corresponding block structure are known. In this paper, we consider the problem of recovering the optimally block-sparsifying block structure for a given data set and dictionary pair. We propose a block structure identification framework employing a clustering step which can be realized using the standard clustering schemes from the literature. The block structure identification algorithm works efficiently, and for synthetically generated block-sparse data the underlying block structure is retrieved even for comparably short data records.

Index Terms— Block-sparsity, dictionary block structure, clustering.

1. INTRODUCTION

Utilizing an overcomplete dictionary to sparsely represent signals leads to efficient representations. The literature is rife with methods developed for sparse decomposition of signals using an overcomplete dictionary formed as a redundant set of basis vectors or atoms. The representation of the signals under consideration in terms of the dictionary atoms might exhibit additional properties other than sparsity. The nonzero coefficients might occur in blocks, in which case the signals will be referred to as block-sparse signals. Block-sparse representations for signals find usage in certain applications, such as multi-band signal reconstruction [1] or sampling of signals from a union of subspaces [2].

Sparse signal representation methods have been extended to block-sparse domain successfully. Attempts at block-sparse representation algorithms include the greedy Block Orthogonal Matching Pursuit (BOMP) [3], combined ℓ_2/ℓ_1 optimization [4, 5] and extensions of iterative sparse coding algorithms to model-based sparse representation [6]. In block-sparse signal representation problem, a dictionary and the corresponding block structure are supplied. The aim is to find the most block-sparse decomposition of the signal under consideration for the given dictionary and block structure pair.

Data driven dictionary design suitable for joint sparse representation of data sets, that is dictionary learning has also garnered considerable attention [7]. There have been some recent attempts at dictionary learning for block-sparse signals [8]. In block-sparse dictionary design the final goal is to find a dictionary and the corresponding block structure suitable for joint block-sparse representation of the observed data set. A common theme underlying both block-sparse signal representation and block-sparse dictionary design is the need for the dictionary block structure which defines how the atoms of the dictionary are grouped together into blocks. The block-sparse signal representation algorithms assume the block structure to be given. A block-sparse dictionary design algorithm on the other hand should generate the block structure together with the evolving dictionary design. Hence, one important step in block-sparse signal processing seems to be the recovery of the underlying block structure for a given block-sparsifying dictionary and data set pair.

In this paper we propose a method for capturing the block structure of the atoms included in a block-sparsifying dictionary. The proposed method utilizes clustering at a certain stage, and we also develop a hierarchical agglomerative clustering algorithm suitable for use at this clustering stage. An algorithm for block structure recovery called as Sparse Agglomerative Clustering (SAC) has been proposed in [8]. The clustering section of SAC comes out to be a special case of the clustering framework proposed in this paper. The clustering framework developed in this paper allows the use of different proximity measures between blocks, and it allows the use of standard clustering algorithms from literature.

2. BLOCK-SPARSE SIGNAL REPRESENTATION

We consider the block-sparse representation of observed signals over a given dictionary and corresponding block structure. The observed signal at each time point n is of dimension M, $\boldsymbol{x}_n \in \mathbb{R}^M$ for $n = 1, \ldots, N$. The dictionary consists of a total of K atoms. Hence, when the atoms $\boldsymbol{d}_i \in \mathbb{R}^M$ are ordered as columns, the dictionary is given as a matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$. The block structure defines how the atoms of the dictionary are sorted in groups or blocks. The distribution of atoms to blocks can be given by an assignment vector $\mathbf{\Gamma} \in \mathbb{R}^K$ [8]. Assuming there are a total of B blocks, $\mathbf{\Gamma}(i) \in \{1, 2, \ldots B\}$ denotes which block the atom $\boldsymbol{d}_i$ belongs to. Each atom is assumed to belong only to a single block, hence block overlaps are not allowed. The set of the indices of atoms included in the j^{th} block of block structure $\mathbf{\Gamma}$ will be denoted by $\mathbf{\Omega}_j^{\mathbf{\Gamma}}$, where $\mathbf{\Omega}_j^{\mathbf{\Gamma}} = \{i | \mathbf{\Gamma}(i) = j\}$. The number of atoms included in block j or the size of block j will be denoted as the cardinality $|\mathbf{\Omega}_j^{\mathbf{\Gamma}}|$.

The representation of a signal vector $\boldsymbol{x}$ over dictionary $\mathbf{D}$ is a vector $\boldsymbol{w} \in \mathbb{R}^K$ such that $\boldsymbol{x} = \mathbf{D}\boldsymbol{w}$. The weight terms corresponding to only block j defined by block structure $\mathbf{\Gamma}$ is given as the subvector $\boldsymbol{w}_j^{\mathbf{\Gamma}} = \{\boldsymbol{w}(i) | i \in \mathbf{\Omega}_j^{\mathbf{\Gamma}}\}$. The block-sparsity of a weight vector $\boldsymbol{w}$ over a block structure defined by $\mathbf{\Gamma}$ is the num-

ber of blocks in which $\boldsymbol{w}$ has non-zero components. Hence, the block-sparsity value of $\boldsymbol{w}$ is equal to the pseudo mixed norm $\|\boldsymbol{w}\|_{2,0}^{\boldsymbol{\Gamma}}$ which counts the number of nonzero blocks in $\boldsymbol{w}$, where $\|\boldsymbol{w}\|_{2,0}^{\boldsymbol{\Gamma}} = \sum_{j=1}^{B} \mathcal{I}\big(\|\boldsymbol{w}_j^{\boldsymbol{\Gamma}}\|_2\big)$. Here, $\mathcal{I}(\cdot)$ is an indicator function, which outputs zero if the argument is zero and one if the argument is non-zero. We will denote the block-sparsity over $\boldsymbol{\Gamma}$ as $\|\boldsymbol{w}\|_{\boldsymbol{\Gamma}} = \|\boldsymbol{w}\|_{2,0}^{\boldsymbol{\Gamma}}$ to make the notation simpler. A vector $\boldsymbol{w}$ is said to be block k-sparse over $\boldsymbol{\Gamma}$, if its non-zero components occur in only k of the total B blocks, hence if $\|\boldsymbol{w}\|_{\boldsymbol{\Gamma}} = \|\boldsymbol{w}\|_{2,0}^{\boldsymbol{\Gamma}} = k$. Using the developed notation, a noise-free formulation for the block-sparse signal representation problem of signal vector $\boldsymbol{x}$ over a dictionary $\mathbf{D}$ and block structure $\boldsymbol{\Gamma}$ can be given as follows.

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|_{\boldsymbol{\Gamma}} \text{ s.t. } \boldsymbol{x} = \mathbf{D}\boldsymbol{w} \tag{1}$$

For the solution of (1), [3] presents a generalization of the greedy Orthogonal Matching Pursuit (OMP) sparse representation algorithm to the block-sparse representation. [4] and [5] both present convex relaxation on the block-sparsity argument à la the Basis Pursuit approach, which utilizes convex relaxation on the ℓ_0 norm in the regular sparse representation problem. The block-sparsity $\|\boldsymbol{w}\|_{\boldsymbol{\Gamma}} = \|\boldsymbol{w}\|_{2,0}^{\boldsymbol{\Gamma}}$ gets replaced with a mixed ℓ_2/ℓ_1 norm [4].

3. BLOCK-SPARSE REPRESENTATION WITH BLOCK STRUCTURE IDENTIFICATION

The block-sparse representation algorithms [3, 4] necessitate the block structure to be given. In this paper we consider the problem of identifying the optimum block-sparsifying block structure and the corresponding block-sparse representation for a given data set and dictionary. This problem is encountered when in block-sparse signal representation the dictionary is known, but the block structure is not available. We define the block-sparse representation with block structure identification problem as follows.

$$\{\hat{\boldsymbol{\Gamma}}, \hat{\mathbf{W}}\} = \underset{\boldsymbol{\Gamma},\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_{\boldsymbol{\Gamma}} \text{ s.t. } \mathbf{X} = \mathbf{D}\mathbf{W} \text{ and } |\boldsymbol{\Omega}_j^{\boldsymbol{\Gamma}}| \leq s, j \in \boldsymbol{\Gamma} \tag{2}$$

Here, $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^{N} \in \mathbb{R}^{M\times N}$ and $\mathbf{W} = \{\boldsymbol{w}_n\}_{n=1}^{N} \in \mathbb{R}^{K\times N}$ are the time concatenated data matrix and the corresponding time concatenated representation matrix, respectively. The argument to minimize is the sum of the block-sparsity values for all $\boldsymbol{w}_n$. The $|\boldsymbol{\Omega}_j^{\boldsymbol{\Gamma}}| \leq s$ condition forces the maximum number of atoms included in each block to be s. Obviously, if we do not put any constraint on $\boldsymbol{\Gamma}$, the most block-sparsifying block structure is $\hat{\boldsymbol{\Gamma}} = [1\ 1 \ldots 1]$, which summons all the atoms in a single block. There should be some constraint on the desired block structure. This constraint might for example be in the form of a required minimum number of blocks or a required maximal block size. We assume that there is a size constraint on the blocks rather than some prior knowledge about the number of groups. Hence, we require a final block structure $\hat{\boldsymbol{\Gamma}}$ with maximal block size s.

This problem is studied as a segment of block-sparse dictionary optimization in [8]. We propose an iterative refinement technique similar to the method proposed in [8], however the clustering section will be different then the method proposed in [8]. We propose a two-step approach for the solution of (2). The approach we propose for the solution of (2) is presented in Alg.1. The first step in Alg.1 initializes the sparse representation as the solution to the regular sparse representation problem. This step can be realized by OMP or BOMP with trivial block structure $\boldsymbol{\Gamma} = [1\ 2 \ldots K]$. The third step in Alg.1 solving (4) is simply the block-sparse representation problem (1) formulated for all time points jointly. This step can be handled by a block-sparse representation solver such as BOMP.

Algorithm 1 Block-Sparse Representation with Block Structure Identification

Input: $\mathbf{D}$, $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^{N}$, and some a priori information on the block structure. We assume maximal block size s is required.

Goal: $\{\hat{\boldsymbol{\Gamma}}, \hat{\mathbf{W}}\} = \underset{\boldsymbol{\Gamma},\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_{\boldsymbol{\Gamma}}$ s.t. $\mathbf{X} = \mathbf{D}\mathbf{W}$ and $|\boldsymbol{\Omega}_j^{\boldsymbol{\Gamma}}| \leq s, j \in \boldsymbol{\Gamma}$.

1: Initialize $\mathbf{W}$ as the solution of the regular sparse representation problem: $\mathbf{W}^{(0)} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_0$ s.t. $\mathbf{X} = \mathbf{D}\mathbf{W}$

2: Find optimally block-sparsifying $\boldsymbol{\Gamma}$ for constant $\mathbf{W}^{(0)}$:

$$\hat{\boldsymbol{\Gamma}} = \underset{\boldsymbol{\Gamma}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\boldsymbol{w}_n^{(0)}\|_{\boldsymbol{\Gamma}} \text{ s.t. } |\boldsymbol{\Omega}_j^{\boldsymbol{\Gamma}}| \leq s, j \in \boldsymbol{\Gamma} \tag{3}$$

3: Find optimally block-sparse $\mathbf{W}$ for constant $\hat{\boldsymbol{\Gamma}}$:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_{\hat{\boldsymbol{\Gamma}}} \text{ s.t. } \mathbf{X} = \mathbf{D}\mathbf{W} \tag{4}$$

4. CLUSTERING BASED FRAMEWORK FOR BLOCK STRUCTURE IDENTIFICATION

The second step (3) of Alg.1 is where the optimally block-sparsifying block structure for a given representation matrix $\mathbf{W}$ should be found. This block structure identification step necessitates the clustering of atoms together into multiple blocks. We propose a general framework for the block structure identification, which can utilize different clustering algorithms in its clustering step. The clustering section of the SAC algorithm proposed in [8] appears to be a special case of this general framework. In the clustering step we allow the use of different metrics which measure similarity between clusters. The use of different metrics helps to achieve better performance in finding the optimally block-sparsifying block structure.

The algorithm we propose for block structure identification is as follows. The row j of $\mathbf{W}$, $\boldsymbol{w}^j \in \mathbb{R}^N$ holds all the information for the usage of the j^{th} atom by the sparse representations $\{\boldsymbol{w}_n\}_{n=1}^{N}$. Atoms which belong to the same block should have nonzero values occurring concurrently in the same positions. Hence, if two atoms j_1 and j_2 belong to the same block, the non-zero values of $\boldsymbol{w}^{j_1}$ and $\boldsymbol{w}^{j_2}$ should occur mostly at the same positions. Let us define what we call as the sparse representation indicator matrix, $\mathbf{I}_W = \mathcal{I}\{\mathbf{W}\}$. $\mathcal{I}\{\cdot\}$ is an indicator function which acts elementwise on the argument matrix. Hence, $\mathbf{I}_W \in \mathbb{R}^{K\times N}$ is equal to zero at the positions where $\mathbf{W}$ is zero and equal to one elsewhere. For the block clustering problem, the usable information for each atom j is contained in the j^{th} row of $\mathbf{I}_W$, $\boldsymbol{i}^j = \mathcal{I}\{\boldsymbol{w}^j\}$. The identification of the block structure for the atoms can be simplified to clustering the rows of $\mathbf{I}_W$ into groups. The general framework we propose for the solution of the block structure identification problem (3) is described in Alg.2.

Block structure identification algorithm as described by Alg.2 necessitates clustering of the K rows of the sparse representation indicator matrix $\mathbf{I}_W$ into groups. The indicator matrix rows $\boldsymbol{i}^j$ are

Algorithm 2 Block Structure Identification

Input: $\mathbf{W} = \{\boldsymbol{w}_n\}_{n=1}^N$, and some a priori information on the block structure such as maximal block size s or total number of blocks B.

Goal: Find optimal $\hat{\mathbf{\Gamma}} = \underset{\mathbf{\Gamma}}{\operatorname{argmin}} \sum_{n=1}^N \|\boldsymbol{w}_n\|_{\mathbf{\Gamma}}$ such that the a priori information on the block structure is valid.

1: Form the sparse representation indicator matrix $\mathbf{I}_W = \mathcal{I}\{\mathbf{W}\}$.

2: Apply clustering algorithm on the rows of $\mathbf{I}_W$, $\{\boldsymbol{i}^j\}_{j=1}^K$. Use the a priori knowledge on the block structure.

3: Form the block structure corresponding to the clustering of the rows of $\mathbf{I}_W$.

binary vectors. The general clustering step for these vectors can employ any one of the many clustering approaches [9]. If one is to apply a hierarchical clustering method, the first decision to make is whether to utilize an agglomerative or partitioning (divisive) approach. In this paper we assume that the a priori knowledge available to the clustering algorithm is the maximum size of the atom blocks, s. Hence, an agglomerative approach is appropriate. A hierarchical agglomerative clustering algorithm initializes with K clusters with a single element or observation in each. At each iteration of the clustering algorithm a similarity or distance matrix is recalculated. The similarity matrix carries the information of the closeness of two clusters. The similarity or closeness between two clusters is calculated by a specified similarity or distance metric, which we will discuss later. After the similarity matrix is generated, the two distinct clusters with the highest similarity value are merged into one single cluster, and the algorithm continues with the next iteration until it is not possible to merge any more clusters without breaking the rule on the maximal cluster size. The steps for a hierarchical agglomerative algorithm which outputs clusters with at most s elements are given in Alg.3.

In Alg.3, the set of the indices of elements included in the m^{th} cluster is denoted by $\mathbf{\Omega}_m$. The number of elements included in the m^{th} cluster is denoted as $|\mathbf{\Omega}_m|$. $\operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\}$ represents the similarity measure between clusters $\mathbf{\Omega}_{m_1}$ and $\mathbf{\Omega}_{m_2}$. There are many possible similarity or distance measures which can be used to quantify the similarity between clusters. In defining a similarity measure between clusters, there are two criteria to determine. One is the similarity metric between two elements or observations, which we will denote by $\mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2})$. The second criterion is how the linkage between clusters is realized. The linkage criterion defines how the similarity between clusters is calculated using the individual similarity metric values between the elements of the two clusters. For the similarity metric $\mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2})$ between the binary observations $\boldsymbol{i}^{j_1}$ and $\boldsymbol{i}^{j_2}$, any one of the many binary similarity measures can be used, ranging from regular Hamming similarity [10] to inner product (IP) metrics [9, 11]. Two possible similarity distance metrics are described below.

Hamming similarity:

$$\mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2}) = \sum_{n=1}^N v_n \text{ where } v_n = \begin{cases} 1, & \boldsymbol{i}^{j_1}[n] = \boldsymbol{i}^{j_2}[n] \\ 0, & \boldsymbol{i}^{j_1}[n] \neq \boldsymbol{i}^{j_2}[n] \end{cases}.$$

Inner Product similarity (IPS):

$$\mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2}) = \sum_{n=1}^N v_n \text{ where } v_n = \begin{cases} 1, & \boldsymbol{i}^{j_1}[n] = \boldsymbol{i}^{j_2}[n] = 1 \\ 0, & \text{otherwise} \end{cases}.$$

Inner product similarity (IPS) counts the number of positions where both binary observation vectors are equal to one. Hence, IPS counts the total number of time instants, at which both atoms are active. The linkage schemes are threefold with the single-link, the complete-link and group average linkage schemes. For a given observation similarity metric $\mathrm{d}(\cdot,\cdot)$, the inter-cluster similarity measures for the three linkage schemes are given as below.

Single-link (SL) linkage:

$$\operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} = \max_{j_1 \in \mathbf{\Omega}_{m_1}, j_2 \in \mathbf{\Omega}_{m_2}} \mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2}).$$

Complete-link (CL) linkage:

$$\operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} = \min_{j_1 \in \mathbf{\Omega}_{m_1}, j_2 \in \mathbf{\Omega}_{m_2}} \mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2}).$$

Group average (GA) linkage:

$$\operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} = \frac{1}{|\mathbf{\Omega}_{m_1}||\mathbf{\Omega}_{m_2}|} \sum_{j_1 \in \mathbf{\Omega}_{m_1}, j_2 \in \mathbf{\Omega}_{m_2}} \mathrm{d}(\boldsymbol{i}^{j_1}, \boldsymbol{i}^{j_2}).$$

Different observation similarity metrics $\mathrm{d}(\cdot,\cdot)$ and different linkage schemes culminate in different cluster similarity measures $\operatorname{sim}\{\cdot,\cdot\}$ which can be used in the HAC-s algorithm given in Alg.3. The resulting HAC-s algorithm is utilized in the step 2 of the Block Structure Identification algorithm given in Alg.2. The SAC algorithm proposed in [8] fits into the block-sparse representation with block structure identification framework presented here. The clustering section of SAC as proposed in [8] can be shown to be in the same form as the HAC-s algorithm with the following inter-cluster similarity measure:

$$\operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} = \vee\{\mathbf{\Omega}_{m_1}\} \cdot \vee\{\mathbf{\Omega}_{m_2}\}^T \tag{5}$$

Here, $\vee\{\cdot\}$ is an *or* operator which acts jointly on the members of the argument set. Hence, $\vee\{\mathbf{\Omega}_m\} = \underset{j \in \mathbf{\Omega}_m}{\vee} \boldsymbol{i}^j$, where $\vee$ is the elementwise *or* operator. For each cluster $\mathbf{\Omega}_m$, SAC saves a single vector which is the elementwise *or* of the member vectors $\boldsymbol{i}^j, \forall j \in \mathbf{\Omega}_m$. As a similarity measure SAC takes the inner product of these vectors. This similarity measure does not correspond to any of the above given binary similarity measures with the specified linkages.

Algorithm 3 Hierarchical Agglomerative Clustering with maximal cluster size s (HAC-s)

Input: K elements to cluster $\{\boldsymbol{i}^j\}_{j=1}^K$ and maximal cluster size s.

Goal: Group the K elements into clusters $\mathbf{\Omega}_m$ such that there are at the most s elements in each cluster.

1: Initialize the clusters as $\mathbf{\Omega}_m \leftarrow \{m\}, m = 1, 2, \ldots, K$.

2: Initialize the similarity matrix as

$$\mathbf{A}(m_1, m_2) = \begin{cases} \operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} & m_1 \neq m_2 \\ 0 & m_1 = m_2 \end{cases}$$

3: **while** $\mathbf{A}$ is not all zeroes matrix **do**

4: Find the two clusters with the maximal similarity value $\max(\mathbf{A})$ and join them into a single cluster. Update the cluster numbering.

5: Update the similarity matrix as

$$\mathbf{A}(m_1, m_2) = \begin{cases} \operatorname{sim}\{\mathbf{\Omega}_{m_1}, \mathbf{\Omega}_{m_2}\} & m_1 \neq m_2 \text{ and } |\mathbf{\Omega}_{m_1}| + |\mathbf{\Omega}_{m_2}| \leq s \\ 0 & m_1 = m_2 \text{ or } |\mathbf{\Omega}_{m_1}| + |\mathbf{\Omega}_{m_2}| > s \end{cases}$$

6: **end while**

5. SIMULATION RESULTS

In this section we inspect the capability of the proposed Block Structure Identification technique (Algs. 1 and 2) in retrieving the underlying block structure from a given dictionary $\mathbf{D}$ and corresponding block-sparse data set $\mathbf{X}$ corrupted with additive noise. $\mathbf{D}$ is a random matrix with independent, identically normal distributed entries, $\mathbf{D} \in \mathbb{R}^{80\times 90}$. $\mathbf{X}$ is a set of N data observations with $\boldsymbol{x}_n \in \mathbb{R}^{80}$. Each data vector is generated as to have a k-block-sparse representation over $\mathbf{D}$. The block structure for $\mathbf{D}$ has 30 blocks with 3 atoms in each. The active blocks for each $\boldsymbol{x}_n$ are chosen randomly, and the coefficients for $\mathbf{W}$ are again random entries. Additive white Gaussian noise is added to the data matrix as to result in desired SNR values. Hence, $\mathbf{X} = \mathbf{DW} + \boldsymbol{\mathcal{N}}$ where $\mathbf{W}$ is the block-sparse weight matrix and $\boldsymbol{\mathcal{N}}$ is the observation noise. For each setting the setup is repeated 50 times. In each iteration, the dictionary $\mathbf{D}$, the block structure $\boldsymbol{\Gamma}$, the block-sparse coefficients $\mathbf{W}$ and hence $\mathbf{X}$ are generated from scratch. The block structure estimate $\hat{\boldsymbol{\Gamma}}$ and the original block structure $\boldsymbol{\Gamma}$ are compared as to calculate the percent of blocks which are identified correctly in $\hat{\boldsymbol{\Gamma}}$.

Fig.1 presents the percent of correctly identified blocks for different block-sparsity values k. The Block Structure Identification procedure as detailed in Alg.2 is realized with HAC-s algorithm employing IP and Hamming similarity metrics with GA linkage scheme. The SAC algorithm [8] is also included for comparison purposes. Each subfigure in Fig.1 presents the results for a specific k value and for three data record lengths $N = 50, 100$ and 200. The plots show that clustering using the IP and Hamming distance metrics result in better block structure identification performance than the SAC algorithm. All algorithms perform 100% recovery for SNR's exceeding a certain threshold. However, before that threshold the HAC-s with IP-GA and Hamming-GA metrics performs better than the SAC. The results for SL and CL linkage are not presented. In our experiments we have observed that they produce results slightly inferior to GA linkage.

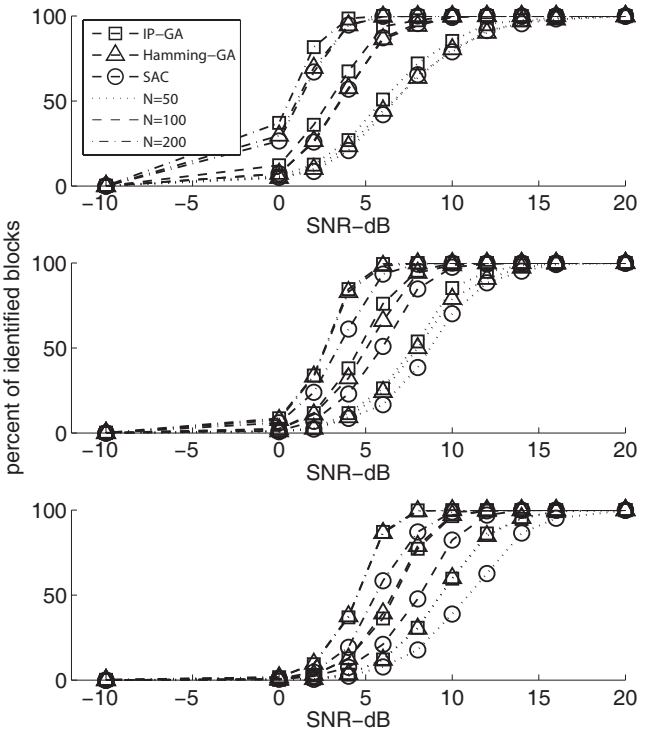

Fig. 1: Percent of correctly identified blocks. The graphs present the block structure identification performance for HAC-s with IP-GA and Hamming-GA similarity measures and the SAC algorithm. a) $k = 4$ b) $k = 6$ c) $k = 8$

6. CONCLUSIONS

We have proposed a framework for the identification of the block structure for block-sparse data. Block-sparse representation algorithms assume the dictionary and the underlying block structure as given. Hence, the retrieval of the block structure from data gains importance. The block structure identification procedure we develop successfully recovers the dictionary block structure even for short data records.

7. REFERENCES

[1] M. Mishali and Y.C. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 993 –1009, Mar. 2009.

[2] T. Blumensath and M.E. Davies, "Sampling theorems for signals from the union of finite-dimensional linear subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1872 –1882, Apr. 2009.

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

[4] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, Aug. 2009.

[5] Y.C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302 –5316, Nov. 2009.

[6] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982 –2001, Apr. 2010.

[7] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2121–2130, Apr. 2010.

[8] K. Rosenblum, L. Zelnik-Manor, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *ArXiv e-prints*, May 2010, [Online]. Available: http://arxiv.org/abs/1005.0202.

[9] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, Wiley, New York, 2. edition, 2001.

[10] J. D. Tubbs, "A note on binary template matching," *Pattern Recogn.*, vol. 22, no. 4, pp. 359–366, 1989.

[11] S. Cha, S. Yoon, and C.C. Tappert, "On binary similarity measures for handwritten character recognition," in *8. International Conf. on Document Analysis and Recognition (ICDAR05)*, Aug. 2005, vol. 1, pp. 4 – 8.