Linear-Time Subspace Clustering via Bipartite Graph Modeling

Amir Adler, Michael Elad, *Fellow, IEEE,* and Yacov Hel-Or,

***Abstract*—We present a linear-time subspace clustering approach that combines sparse representations and bipartite graph modeling. The signals are modeled as drawn from a union of low dimensional subspaces, and each signal is represented by a sparse combination of basis elements, termed *atoms*, which form the columns of a dictionary matrix. The sparse representation coefficients are arranged in a sparse affinity matrix, which defines a bipartite graph of two disjoint sets: atoms and signals. Subspace clustering is obtained by applying low-complexity spectral bipartite graph clustering that exploits the small number of atoms for complexity reduction. The complexity of the proposed approach is linear in the number of signals, thus, it can rapidly cluster very large data collections. Performance evaluation of face clustering and temporal video segmentation demonstrate comparable clustering accuracies to state-of-the-art at a significantly lower computational load.**

***Index Terms*—subspace clustering, dictionary, sparse representation, bipartite graph, face clustering, temporal video segmentation.**

I. Introduction

Dimensionality reduction is a powerful tool for processing high dimensional data such as video, image, audio and biomedical signals. The simplest of such techniques is Principal Component Analysis (PCA) that models the data as spanned by a single low-dimensional subspace, however, in many cases a *union-of-subspaces* model can represent more accurately the data: for example [1] proposed to generalize PCA to identify multiple subspaces for computer vision applications, [2] proposed to generalize k-means to cluster facial images and [3] proposed efficient sampling techniques for practical signal types that emerge from a union-of-subspaces model. Subspace clustering is the problem of clustering a collection of signals drawn from a union-of-subspaces, according to their spanning subspaces. Subspace clustering algorithms can be divided into four approaches: statistical, algebraic, iterative and spectral clustering-based; see [4] for a review. State-of-the-art approaches such as Sparse Subspace Clustering [5], [6], Low-Rank Representation [7], [8] and Low-Rank Subspace Clustering [9] are spectral-clustering based. These methods provide excellent performance, however, their complexity limits the size of the data sets to $\approx 10^4$ signals. K-subspaces [2] is a generalization of the K-means algorithm to subspace clustering that can handle large data sets, however, it requires explicit knowledge of the dimensions of all subspaces and its performance is inferior compared to state-of-the-art. In this paper we address the problem of applying subspace clustering to very large data collections. This problem is important due to the following reasons: 1) Existing subspace clustering tasks are required to handle the ever-increasing amounts of data such as image and video streams. 2) Subspace clustering based solutions could be applied to applications that traditionally could not employ subspace clustering, and require large data processing.

In the following we formulate the subspace clustering problem, review previous works based on sparse and low-rank modeling and highlight the properties of our approach.

A. Adler and M. Elad are with the Department of Computer Science, Technion, Haifa 32000, Israel. Y. Hel-Or is with the Department of Computer Science, The Interdisciplinary Center, Herzlia 46150 , Israel.

A. Problem Formulation

Let $\mathbf{Y} \in \mathbb{R}^{N \times L}$ be a collection of L signals $\{\mathbf{y}_l \in \mathbb{R}^N\}_{l=1}^{L}$, drawn from a union of $K > 1$ linear subspaces $\{\mathbf{S}_i\}_{i=1}^{K}$. The bases of the subspaces are $\{\mathbf{B}_i \in \mathbb{R}^{N \times d_i}\}_{i=1}^{K}$ and $\{d_i\}_{i=1}^{K}$ are their dimensions. The task of subspace clustering is to cluster the signals according to their subspaces. The number of subspaces K is either assumed known or estimated during the clustering process. The difficulty of the problem depends on the following parameters:

1) Subspaces separation: the subspaces may be independent (as defined in Appendix A), disjoint or some of them may intersect, which is considered the most difficult case.
2) Signal quality: the collection of signals Y may be corrupted by noise, missing entries or outliers, thus, distorting the true subspaces structure.
3) Model Accuracy: the union-of-subspaces model is often only an approximation of a more complex and unknown data generation model, and the magnitude of the error it induces affects the overall performance.

B. Prior Art: Sparse and Low Rank Modeling

Sparse Subspace Clustering (SSC) and Low-Rank Representation (LRR) reveal the relations among signals by finding a self-expressive representation matrix $\mathbf{W} \in \mathbb{R}^{L \times L}$ such that $\mathbf{Y} \simeq \mathbf{YW}$, and obtain subspace clustering by applying spectral clustering [10] to the graph induced by $\mathbf{W}$. SSC forces $\mathbf{W}$ to be sparse by solving the following set of optimization problems, for the case of signals contaminated by noise with standard deviation ε (section 3.3 in [5]):

$$\min_{\mathbf{w}_i} \|\mathbf{w}_i\|_1 \text{ s.t. } \|\mathbf{Y}_{\hat{i}}\mathbf{w}_i - \mathbf{y}_i\|_2 \leq \varepsilon \text{ (for i=1 } \cdots \text{ L)}, \qquad (1)$$

where $\mathbf{w}_i \in \mathbb{R}^{L-1}$ is the sparse representation vector, $\mathbf{y}_i$ is the i-th signal and $\mathbf{Y}_{\hat{i}}$ is the signal matrix $\mathbf{Y}$ excluding the i-th signal. By inserting a zero at the i-th entry of $\mathbf{w}_i$ and augmenting the dimension of $\mathbf{w}_i$ to L, the vector $\hat{\mathbf{w}}_i \in \mathbb{R}^L$ is obtained, which defines the i-th column of $\mathbf{W} \in \mathbb{R}^{L \times L}$, such that diag($\mathbf{W}$)=0.

For the case of signals with sparse outlying entries, SSC forces $\mathbf{W}$ to be sparse by solving the following optimization problem:

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_1 + \lambda \|\mathbf{E}\|_1 \text{ s.t. } \mathbf{Y} = \mathbf{YW} + \mathbf{E} \text{ and } \text{diag}(\mathbf{W}) = 0, \quad (2)$$

where E is a sparse matrix representing the sparse errors in the data and $\lambda > 0$. LRR forces $\mathbf{W}$ to be low-rank by minimizing its nuclear norm (sum of singular values), and solves the following optimization problem for clustering signals contaminated by noise and outliers:

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_* + \lambda \|\mathbf{E}\|_{2,1} \text{ s.t. } \mathbf{Y} = \mathbf{YW} + \mathbf{E}. \quad (3)$$

SSC was reported to outperform Agglomerative Lossy Compression [11] and RANSAC [12], whereas LRR was reported to outperform Local Subspace Affinity [13] and Generalized-PCA [1]. LRR and SSC provide excellent performance, however, they are restricted to relatively moderate-sized data sets due the following reasons:

1) Polynomial complexity affinity calculation - SSC solves L sparse coding problems with a dictionary of $L-1$ columns, leading to approximate complexity of $O(L^2)$. The complexity of LRR is higher as its Augmented Lagrangian-based solution involves repeated SVD computations of an $L \times L$ matrix during the convergence to W, leading to complexity of $O(L^3)$ multiplied by the number of iterations (which can exceed 100).
2) Polynomial complexity spectral clustering - both LRR and SSC require eigenvalue decomposition (EVD) of an $L \times L$ Laplacian matrix, leading to polynomial complexity of the spectral clustering stage[1]. In addition, the memory space required to store the entries of the graph Laplacian is $O(L^2)$, which becomes prohibitively large for $L \gg 1$.

In addition, whenever the entire data set is contaminated by noise, both LRR and SSC suffer from degraded performance since each signal in Y is represented by a linear combination of other **noisy** signals. Low-rank subspace clustering (LR-SC) [9] provides closed-form solutions for noisy data and iterative algorithms for data with outliers. LR-SC provides solutions for noisy data by introducing the clean data matrix Q and solving relaxations of the following problem:

$$\min_{\mathbf{W},\mathbf{E},\mathbf{Q}} \|\mathbf{W}\|_* + \lambda \|\mathbf{E}\|_F \text{ s.t. } \mathbf{Q} = \mathbf{QW} \text{ and } \mathbf{Y} = \mathbf{Q} + \mathbf{E}. \quad (4)$$

Note that the computational load of the spectral clustering stage remains the same as that of LRR and SSC, since the dimensions of the affinity matrix remains $L \times L$. The clustering accuracy of LR-SC was reported as comparable to SSC and LRR, while better than Shape Interaction Matrix [14], Agglomerative Lossy Compression [11] and Local Subspace Affinity [13]. The work of [15] proposed a dictionary-based approach that learns a set of K sub-dictionaries (for K data classes) using a Lloyd's-type algorithm that is initialized by applying spectral clustering to a graph of atoms or a graph of signals. Each signal is assigned to a class according to the sub-dictionary that best represents it, using a novel metric defined in this work. The work of [16] proposed a dictionary-based approach, which employs a probabilistic mixture model to compute signals likelihoods and obtains subspace clustering using a maximum-likelihood rule.

[1]Note that a full EVD of the Laplacian has complexity of $O(L^3)$, however, a complexity of $O(L^2)$ is required for computing only several eigenvectors.

C. Paper Contributions

This paper presents a new spectral clustering-based approach that is built on sparsely representing the given signals using a dictionary, which is either learned or known a-priori[2]. The contributions of this paper are as follows:

1) Bipartite graph modeling: a novel solution to the subspace clustering problem is obtained by mapping the sparse representation matrix to an affinity matrix that defines a bipartite graph with two disjoint sets of vertices: dictionary atoms and signals.
2) Linear-time complexity: the proposed approach exploits the small number of atoms M for complexity reduction, leading to an overall complexity that depends only linearly on the number of signals L.
3) Theoretical study: the conditions for correct clustering of independent subspaces are proved for the cases of minimal and redundant dictionaries.

This paper is organized as follows: Section II overviews sparse representations modeling, which forms the core for learning the relations between signals and atoms. Section III presents bipartite graphs and the proposed approach. Section IV provides performance evaluation of the proposed approach and compares it to leading subspace clustering algorithms.

II. Sparse Representations Modeling

Sparse representations provide a natural model for signals that live in a union of low dimensional subspaces. This modeling assumes that a signal $\mathbf{y} \in \mathbb{R}^N$ can be described as $\mathbf{y} \simeq D\mathbf{c}$, where $\mathbf{D} \in \mathbb{R}^{N \times M}$ is a *dictionary* matrix and $\mathbf{c} \in \mathbb{R}^M$ is sparse. Therefore, $\mathbf{y}$ is represented by a linear combination of a *few* columns (atoms) of $\mathbf{D}$. The recovery of $\mathbf{c}$ can be cast as an optimization problem:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{Dc}\|_2 \leq \varepsilon, \quad (5)$$

for some approximation error threshold ε. The l_0 norm $\|\mathbf{c}\|_0$ counts the non-zeros components of $\mathbf{c}$, leading to a NP-hard problem. Therefore, a direct solution of (5) is infeasible. An approximate solution is given by applying the OMP algorithm [19], which successively approximates the sparsest solution. The recovery of $\mathbf{c}$ can be cast also by an alternative optimization problem that limits the cardinality of $\mathbf{c}$:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{y} - \mathbf{Dc}\|_2 \text{ s.t. } \|\mathbf{c}\|_0 \leq T_0, \quad (6)$$

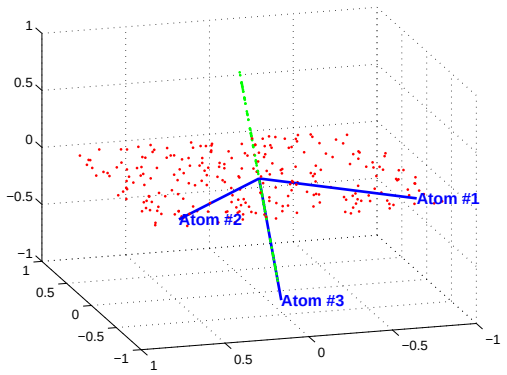
where T_0 is the maximum cardinality. The dictionary $\mathbf{D}$ can be either predefined or learned from the given set of signals, see [20] for a review. For example, the K-SVD algorithm

[2]For example over-complete DCT-based dictionaries are well suited for sparsely representing image patches [17] or audio frames [18].

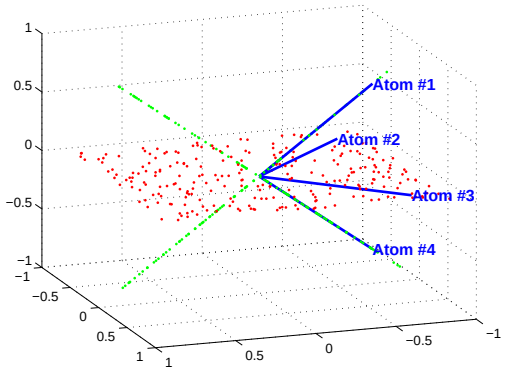
[21] learns a dictionary by solving the following optimization problem:

$$\{\mathbf{D},\mathbf{C}\} = \arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{Y}-\mathbf{D}\mathbf{C}\|_F^2 \ \text{ s.t. } \forall i \|\mathbf{c}_i\|_0 \leq T_0, \qquad (7)$$

where $\mathbf{Y} \in \mathbb{R}^{N\times L}$ is the signals matrix, containing $\mathbf{y}_i$ in it's i-th column. $\mathbf{C} \in \mathbb{R}^{M\times L}$ is the sparse representation matrix, containing the sparse representation vector $\mathbf{c}_i$ in it's i-th column. Once the dictionary is learned, each one of the signals $\{\mathbf{y}_i\}_{i=1}^L$ is represented by a linear combination of few atoms. Each combination of atoms defines a low dimensional subspace, thus, our subspace clustering approach exploits the fact that signals spanned by the same subspace are represented by similar groups of atoms. In the following, we demonstrate this property for signals that are drawn from a union of independent or disjoint subspaces. Consider data points drawn from a union of two independent subspaces in $\mathbb{R}^3$: a plane and a line, as illustrated in Fig. 1(a). A dictionary with 3 atoms was learned from few hundreds of such points, using the K-SVD algorithm, and as illustrated in Fig. 1(a) the learned atoms coincide with the correct bases of the two subspaces. Next, consider data points drawn from a union of three disjoint subspaces in $\mathbb{R}^3$: a plane and two lines, as illustrated in Fig. 1(b). A dictionary with 4 atoms was learned from few hundreds of such points, using the K-SVD algorithm, and as depicted in Fig. 1(b) the learned atoms coincide with the correct bases of the three subspaces.



(a)



(b)

Fig. 1: Dictionary learning of (a) independent and (b) disjoint subspaces' bases.

III. The Proposed Approach

A. From Bipartite Graphs to Subspace Clustering

The sparse representations matrix C provides explicit information on the relations between signals and atoms, which we leverage to quantify the latent relations among the signals: the locations of non-zero coefficients in C determine the atoms that represent each signal and their absolute values determine the respective weights of the atoms in each representation. Therefore, subspace clustering can be obtained by a *bi-clustering* approach: simultaneously grouping signals with the atoms that represent them, such that a cluster label is assigned to every signal and every atom, and the labels of the signals provide the subspace clustering result. In cases where a partition into disjoint groups does not exist (as a result of intersecting subspaces, errors in the sparse coding stage or noise), a possible approach is to group together signals with the most significant atoms that represent them. This *bi-clustering* problem can be solved by bipartite graph partitioning [22]: let $G = (\mathcal{D},\mathcal{Y},E)$ be an undirected bipartite graph consisting of two disjoint sets of vertices: atoms $\mathcal{D} = \{\mathbf{d}_1,\mathbf{d}_2,...,\mathbf{d}_M\}$ and signals $\mathcal{Y} = \{\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_L\}$, connected by the corresponding set of edges E. An edge between an atom and a signal exists only if the atom is part of the representation of the signal. The two disjoint sets of vertices are enumerated from 1 to $M+L$: the leading M vertices are atoms and the tailing L vertices are signals, as illustrated in Fig. 2(a). Let $\mathbf{W} = \{w_{ij}\}$ be a non-negative affinity matrix, such that every pair of vertices is assigned a weight w_{ij}. The affinity matrix is defined by:

$$\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)}, \qquad (8)$$

where $\mathbf{A} = |\mathbf{C}|$. Note that the structure of $\mathbf{W}$ implies that only signal-atom pairs can be assigned a positive weight (in cases the atom is part of the representation of the signal). The matrix $\mathbf{W}$ is used to define the set of edges, such that an edge between the i-th and j-th vertices exists in the graph only if $w_{ij} > 0$ and the weight of this edge is $e_{ij} = w_{ij}$. Thus, the unique structure of $\mathbf{W}$ imposes only one type of connected components: bipartite components that are composed of at least one atom and one signal. This type of graph modeling differs from the modeling employed by LRR, SSC and LRSC, since these methods construct a graph with only a single type of vertices (which are signals) and seek for groups of connected signals. In addition, bipartite graph modeling differs from the method of [15] that partitions either a graph of atoms or a graph signals (each graph with only a single type of vertices), as an initialization stage of the K sub-dictionaries learning algorithm.

A reasonable criterion for partitioning the bipartite graph is the Normalized-Cut [10], which seeks well separated groups while balancing the size of each group, as illustrated in Fig. 2(b). Let $\mathcal{V}_1,\mathcal{V}_2$ be a partition of the graph such that $\mathcal{V}_1 = \mathcal{D}_1 \cup \mathcal{Y}_1$ and $\mathcal{V}_2 = \mathcal{D}_2 \cup \mathcal{Y}_2$, where $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}$ and $\mathcal{Y}_1 \cup \mathcal{Y}_2 = \mathcal{Y}$. The Normalized-Cut partition is obtained by minimizing the

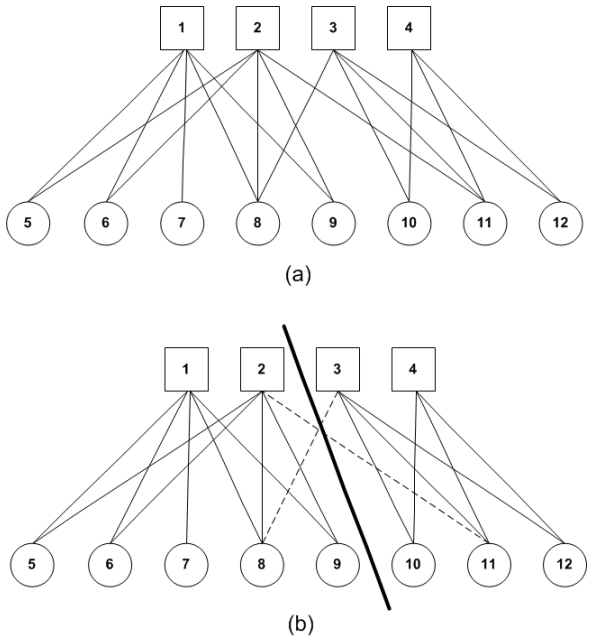


Fig. 2: a) A bipartite graph consisting of 12 vertices: 4 atoms (squares) and 8 signals (circles). b) The signals were drawn from a union of two subspaces, however, the sparse coding stage (OMP) produced inseparable groups in the graph. The Normalized Cut approach attempts to resolve this by grouping together signals with the atoms that are the most significant in the signals' representations. The edges that correspond to the least significant links between atoms to signals are neglected (dashed edges in the figure). The graph partitioning solution is illustrated by the bold line: the vertices of the first group are {1,2,5,6,7,8,9} and the vertices of the second group are {3,4,10,11,12}.

following expression:

$$N_{cut}(\mathcal{V}_1,\mathcal{V}_2)=\frac{\text{cut}(\mathcal{V}_1,\mathcal{V}_2)}{\text{weight}(\mathcal{V}_1)}+\frac{\text{cut}(\mathcal{V}_1,\mathcal{V}_2)}{\text{weight}(\mathcal{V}_2)}, \tag{9}$$

where $\text{cut}(\mathcal{V}_1,\mathcal{V}_2)=\sum_{i\in\mathcal{V}_1,j\in\mathcal{V}_2}W_{ij}$ quantifies the accumulated edge weights between the groups and $\text{weight}(\mathcal{V})=\sum_{i\in\mathcal{V}}\sum_k W_{ik}$ quantifies the accumulated edge weights within a group. Therefore, we propose to partition the bipartite graph using the Normalized-Cut criterion, and obtain subspace clustering from the signals' cluster labels.

Direct minimization of (9) leads to an NP-hard problem, therefore, spectral clustering [10] is often employed as an approximate solution to this problem. A low complexity bipartite spectral clustering algorithm was derived in [22] for natural language processing applications. This algorithm is detailed in Appendix B, and requires the SVD of an $M\times L$ matrix which has complexity of $O(M^2L)$ [23]. Note that in our modeling the number of atoms is fixed and obeys $M\ll L$, leading to complexity that depends linearly in L (compared to the complexity of the spectral clustering stage of state-of-the-art approaches [6], [8], [9] that is polynomial is L). We leverage the SVD-based algorithm to our problem and incorporate it into the proposed algorithm, as detailed in Algorithm 1. The overall complexity of the proposed approach depends only linearly on L, and is given by $O(qJNML)+O(qNML)+O(M^2L)+O(TNKL)$, where the first term is K-SVD complexity (with J iterations and assuming $L\gg 1$), the second term is OMP complexity, the third (SVD complexity) and forth (k-means complexity with T iterations) terms compose the bipartite spectral clustering stage complexity.

Algorithm 1 Subspace Bi-Clustering (SBC)

Input: data $\mathbf{Y}\in\mathbb{R}^{N\times L}$, # of clusters K, # of atoms M.

1. **Dictionary Learning:** Employ K-SVD to learn a dictionary $\mathbf{D}\in\mathbb{R}^{N\times M}$ from $\mathbf{Y}$.
2. **Sparse Coding:** Construct the sparse matrix $\mathbf{C}\in\mathbb{R}^{M\times L}$ by the OMP algorithm, such that $\mathbf{Y}\simeq\mathbf{DC}$.
3. **Bi-Clustering:**
   I. Construct the matrix $\mathbf{A}=|\mathbf{C}|$.
   II. Compute the rank-M SVD of $\overline{\mathbf{A}}=\mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ as in equation (11).
   III. Construct the matrix $\mathbf{Z}=\begin{bmatrix}\mathbf{D}_1^{-\frac{1}{2}}\mathbf{U}\\ \mathbf{D}_2^{-\frac{1}{2}}\mathbf{V}\end{bmatrix}$, where $\mathbf{U}=[\mathbf{u}_2...\mathbf{u}_K]$ and $\mathbf{V}=[\mathbf{v}_2...\mathbf{v}_K]$ as in equation (17). The M leading rows of $\mathbf{Z}$ correspond the atoms and the L tailing rows correspond the signals.
   IV. Cluster the rows of $\mathbf{Z}$ using k-means.

Output: cluster labels for all signals $\hat{k}(y_j), j=1..L$.

B. Theoretical Study

In the following we provide two theorems that pose conditions for correct segmentation of independent subspaces using the proposed approach. Our analysis proves that given a correct dictionary, OMP will always recover successfully the bipartite affinity matrix[3]. Further segmentation of the bipartite graph using the normalized-cut criterion leads to correct subspace clustering. The following theorem addresses the case of a dictionary $\mathbf{D}$ that contains the set of minimal bases for all subspaces:

Theorem 1. *Let $\mathbf{Y}=[\mathbf{Y}_1,\mathbf{Y}_2,\cdots,\mathbf{Y}_K]$ be a collection of $L=L_1+L_2+\cdots+L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^K$. Given a dictionary $\mathbf{D}=[\mathbf{D}_1,\mathbf{D}_2,\cdots,\mathbf{D}_K]$ such that $\mathbf{D}_i\in\mathbb{R}^{N\times d_i}$ spans $\mathbf{S}_i$ and $d_i=dim(\mathbf{S}_i)$, OMP is guaranteed to recover the correct and unique sparse representations matrix $\mathbf{C}$ such that $\mathbf{Y}=\mathbf{DC}$, and minimization of the Normalized-Cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.*

The proof is provided in Appendix C.

We now address the more general case of a redundant dictionary in which the sub-dictionaries are redundant $\mathbf{D}_i\in\mathbb{R}^{N\times t_i}$ and $t_i>d_i$. This situation is realistic in dictionary learning, whenever the number of allocated atoms is higher than necessary. Note that for a redundant dictionary, there is an infinite number of exact representations for each signal $\mathbf{y}_i\in\mathbf{S}_i$, and

[3]Note that this statement is far stronger than a successful OMP conditioned on RIP [24] or mutual-coherence [25], since (1) we address the case of independent subspaces; and (2) our goal is segmentation and not signal recovery.

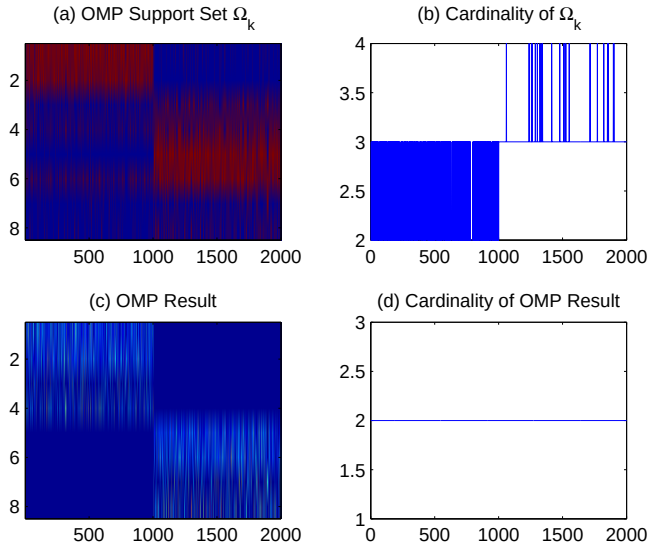


Fig. 3: Sparse representation recovery using OMP with a redundant dictionary and a data collection $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2] \in \mathbb{R}^{4\times 2000}$, where $\mathbf{Y}_{1,2} \in \mathbb{R}^{4\times 1000}$ are drawn from two independent subspaces of dimension 2 each. A redundant dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2] \in \mathbb{R}^{4\times 8}$, with 4 atoms per subspace was used to compute the sparse representation of each data point: (a) the recovered support set often contains atoms of the wrong subspace. (b) The cardinality of the support set often exceeds the correct dimension of 2. Owing to the pseudo-inverse in the OMP operation, the wrong coefficients are effectively nulled, thus leading to (c) perfectly correct supports, and (d) correct cardinalities.

OMP is prone to select wrong atoms (that represent subspaces $\mathbf{S}_j \neq \mathbf{S}_i$) during its operation. However, the following theorem proves that the support of the OMP solution is guaranteed to include atoms **only** from the correct subspace basis (although the accumulated support-set might contain atoms that represent other subspaces). Figure 3 demonstrate this in practice.

Theorem 2. *Let* $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_K]$ *be a collection of* $L = L_1 + L_2 + \cdots + L_K$ *signals from K independent subspaces of dimensions* $\{d_i\}_{i=1}^K$. *Given a dictionary* $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$ *such that* $\mathbf{D}_i \in \mathbb{R}^{N\times d_i}$ *spans* $\mathbf{S}_i$ *and* $d_i > dim(\mathbf{S}_i)$, *OMP is guaranteed to recover a correct sparse representations matrix* $\mathbf{C}$ *such that* $\mathbf{Y} = \mathbf{DC}$, $\mathbf{C}$ *include* only *atoms from the correct subspace basis for each signal, and minimization of the Normalized-Cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.*

The proof is provided in Appendix C.

The next natural steps in studying the theoretical properties and limitations of our proposed scheme are to explore more general cases of disjoint subspaces instead of independent ones, and also explore the sensitivity to a wrong dictionary. We choose to leave these important questions for future work.

IV. Performance Evaluation

This section evaluates[4] the performance of the proposed approach for synthetic data clustering, face clustering and temporal video segmentation. In addition, the performance of SSC, LRR, LR-SC, PSSC [16] and K-subspaces are compared, using code packages that were provided by their authors (the parameters of all methods were optimized for best perfomance). The objective of this section is to demonstrate that as long as the collection size L is sufficiently large for training the dictionary, then the clustering accuracy of the proposed approach is comparable to state-of-the-art algorithms. The correct number of clusters was supplied to all algorithms in every experiment. All experiments were conducted using a computer with Intel *i*7 Quad Core 2.2GHz and 8GB RAM.

A. Computation Time

Computation time comparison of clustering L signals (up-to $L = 1,048,576$) in $\mathbb{R}^{64}$ is provided in Fig. 4. The reported durations for the proposed approach include a dictionary $D \in \mathbb{R}^{64\times 64}$ learning stage from the L signals if $L < 2^{15}$ or 2^{15} signals if $L \geq 2^{15}$. The results indicate polynomial complexity in L of state-of-the-art approaches compared to linear complexity in L of K-subspaces and the proposed approach.

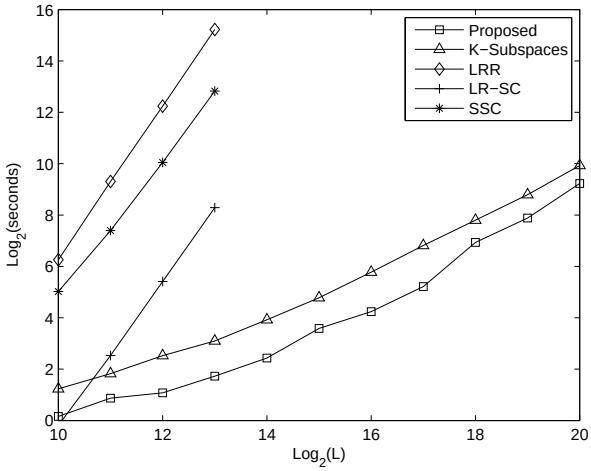


Fig. 4: Computation time vs. number of data samples L, for $K = 32$ subspaces, data samples dimension $N = 64$ and $M = 64$ learned atoms.

B. Synthetic Data Clustering

Clustering accuracy[5] was evaluated for signals contaminated by zero mean white Gaussian noise, in the Signal-to-Noise (SNR) range of 5dB to 20dB. Per each experiment we generated a set of 800 signals in $\mathbb{R}^{100}$ drawn from a union of 8 subspaces of dimensions 10, with equal number of signals per subspace. The bases of all subspaces were chosen as random combinations (non-overlapping for disjoint subspaces) of the columns of a 100×200 Over-complete Discrete Cosine Transform (ODCT) matrix [21]. The coefficients of each signal were randomly sampled from a Gaussian distribution of zero mean and unit variance. Clustering accuracy results, averaged over 10 noise realizations per SNR point, are presented in

[4] All the results presented in the paper are reproducible using a MATLAB package that is freely available for distribution.

[5] Accuracy was computed by considering all possible permutations and define by: $1 - \frac{\text{number of miss-classified signals}}{\text{total number of signals}}$.

Table I. The results of the proposed approach (SBC) are based on a learned dictionary $D \in \mathbb{R}^{100\times 100}$ per every noise realization. The results demonstrate comparable clustering accuracies of the proposed approach and state-of-the-art[6], and superior performance compared to K-Subspaces. Note that (only) the results of K-Subspaces are based on explicit knowledge of the true dimensions ($d = 10$) of all subspaces, as this parameter is required by K-Subspaces. For the proposed approach we employed OMP to approximate the solution of equation (5) and set the sparse representation target error ε to the noise standard deviation (the target error used for SSC was also close to the noise standard deviation).

TABLE I: Clustering accuracy (%) for L=800 signals in $\mathbb{R}^{100}$ drawn from 8 disjoint subspaces with dimension 10: mean, median, standard deviation with respect to mean (σ_{Mean}) and to median (σ_{Median}).

SNR	Params.	SBC	LR-SC	SSC	LRR	K-Sub.
5dB	Mean	**99.9**	97.64	85.47	89.03	82.96
	Median	**100**	99.19	82.00	82.13	87.12
	σ_{Mean}	**0.04**	1.47	2.30	2.82	5.01
	σ_{Median}	**0.05**	1.55	2.55	3.57	5.18
10dB	Mean	**99.97**	99.00	85.53	89.29	87.38
	Median	**100**	99.38	82.13	83.00	92.37
	σ_{Mean}	**0.02**	0.29	4.26	2.77	5.06
	σ_{Median}	**0.02**	0.32	4.39	3.41	5.24
15dB	Mean	98.65	**99.01**	87.42	89.44	97.08
	Median	**100**	99.19	82.44	83.13	100
	σ_{Mean}	1.25	**0.25**	2.61	2.73	1.61
	σ_{Median}	1.33	**0.25**	3.05	3.38	1.86
20dB	Mean	**99.93**	99.06	89.24	90.90	96.02
	Median	99.94	99.25	82.50	91.31	**100**
	σ_{Mean}	**0.03**	0.23	2.78	2.88	1.93
	σ_{Median}	**0.03**	0.24	3.50	2.88	2.30

C. Face Clustering

Face clustering is the problem of clustering a collection of facial images according to their human identity. Facial images taken from the same view-point and under varying illumination conditions are well approximated as spanned by a subspace of dimension < 10 [26], [27], where a unique subspace is associated with each view point and human subject. Subspace clustering was applied successfully to this problem for example in [2], [7]. Face clustering accuracy was evaluated using the Extended Yale B database [28], which contains 16128 images of 28 human subjects under 9 view-points and 64 illumination conditions (per view-point). In our experiments we allocated 10 atoms per human subject (assuming each subspace dimension < 10), and in order to enable efficient dictionary training we found that a minimum ratio of $L/M > 10$ is required for good clustering results (i.e. at least a hundred facial images per subject). Therefore, we generated from the complete collection a subset of 1280 images containing the first 10 human subjects, with 128 images per subject, by merging the 4th and 5th view-points which are of similar angles. We further verified that the 4th and 5th view-points (of each human subject) can be modeled using a single subspace, by reconstructing all 128 images from their projections onto their 9 leading PCA basis vectors (obtained by the PCA of each merged class of 128 images). Visual results of this procedure are provided for the third human subject in Fig. 5, demonstrating excellent quality of the reconstructed images. All images were cropped, resized to 48×42 pixels and column-stacked to vectors in $\mathbb{R}^{2016}$. Clustering accuracy was evaluated for $K = 2..8$ classes, by averaging clustering results over 40 different subsets of human subjects, for each value of K, by choosing 40 different combinations of human subjects out of the 10 classes. Clustering results, provided in Table II, indicate comparable accuracies of the proposed approach to state-of-the-art[7] and consistent advantage compared to PSSC [16] and K-Subspaces. The parameters of each method were optimized for best performance and summarized in Table III. For the proposed approach we employed OMP to approximate the solution of equation (6) and set $T_0 = 9$. We also noticed that many entries of $|C|$ are below 1 whereas few are above 1, and a small clustering accuracy advantage can be obtained by computing the affinity matrix (8) using $A = |C|^p$ with $0 < p < 1$ (rather than $p = 1$). This balances edges' weights by increasing values below 1 and decreasing values above 1 ($p = 0.4$ provided the best results). Note that a similar approach was suggested by [8] in section 5.4 with $p > 1$.

Fig. 5: Reconstruction of facial images from the 3rd merged class of the Extended Yale B collection (the 5 leftmost columns are from the 4th view point and the 5 rightmost columns are from the 5th view point): the first row displays the original images and the second row displays the reconstructed images from their projections onto the 9 leading PCA basis vectors, as obtained from the PCA of the 128 images in the merged class (the union of the 4th and 5th view points).

TABLE II: Face clustering accuracy (%), averaged over 40 different human subjects combinations per each number of clusters (K).

K =	2	3	4	5	6	7	8
Proposed	92.26	**91.03**	**89.13**	**83.42**	72.15	67.07	64.19
LRR	93.75	85.94	65.47	57.02	51.34	52.86	54.88
LRR-H	91.88	72.36	76.36	74.91	72.49	68.19	66.29
SSC	**95.57**	89.11	85.44	78.98	**73.16**	**72.59**	**73.36**
LR-SC	94.51	80.98	72.86	67.11	59.08	58.82	55.53
PSSC	92.19	82.07	78.26	68.77	61.97	56.96	50.58
K-Subs.	67.60	59.17	50.24	51.34	48.81	48.32	45.35

[6]SSC was evaluated using the code that solves equation (1) with ε = noise standard deviation (as defined in section 3.3 of [5]), LRR ($\lambda = 0.15$) was evaluated using the code that solves equation (3) and LR-SC ($\tau = 0.01$) was evaluated using the code that solves Lemma 1 in [9].

[7]State-of-the-art methods were evaluated with sparse outliers support: SSC with the ADMM-based version that solves equation (2), LRR with the version that solves equation (3), LRR-H same as LRR but with post-processing of the affinity matrix [8] and LR-SC with the version that solves equation (4).

TABLE III: Face clustering: algorithms parameters settings.

K =	2	3	4	5	6	7	8
Proposed, $M=$	20	30	40	50	60	70	80
LRR, $\lambda=$	0.25	0.25	0.25	0.3	0.3	0.35	0.35
LR-SC, $(\tau,\gamma)=$	(5,5)	(6,3)	(6,4)	(6,4)	(6,4)	(6,4)	(6,4)
SSC, $(\rho,\alpha)=$	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)
K-Subs., $d=$	8	8	8	8	8	8	8

TABLE IV: Temporal video segmentation accuracy (%) for two sequences: V_1 (1190 frames from ABC's TV show "Wheel Of Fortune") and V_2 (12000 frames from ABC's TV show "One Plus One").

Method	Accuracy (V_1)	Accuracy (V_2)
Proposed	98.99	99.41
SSC	97.82	N/A
LRR-H	99.16	N/A
LR-SC	98.91	N/A

D. Temporal Video Segmentation

Temporal video segmentation is the problem of clustering the frames of a video sequence according to the scene each belongs to (the same scene may repeat several times). By modeling each frame as a point in a high-dimensional linear space, and each scene as spanned by a low-dimensional subspace, temporal video segmentation was successfully solved using subspace clustering in [1]. This work employed GPCA to segment short video sequences of up to 60 frames. In our experiments we evaluated segmentation accuray and computational load for two video sequences. The first sequence V_1 contained 6 scenes and 1190 frames (30 frames-per-second) of dimensions 360 x 640 pixels in RGB format. The frames of V_1 were converted to gray-scale, down-sampled to 90 x 160 pixels and column stacked to vectors in $\mathbb{R}^{14400}$. The second sequence V_2 contained 3 scenes and 12000 frames (25 frames-per-second) of dimensions 288 x 512 pixels in RGB format. The frames of V_2 were converted to gray-scale, down-sampled to 72 x 128 pixels and column stacked to vectors in $\mathbb{R}^{9216}$. In order to determine the number of dictionary atoms, we computed the PCA basis of several scenes (for each one separately) and found that $\sim 80\%$ of the energy of each scene is represented by its 9 leading PCA basis vectors. Therefore, we allocated $9 \times K$ atoms (K is the number of scenes) for the dictionary of each video sequence. The correct segmentation of both sequences was obtained manually, and segmentation accuracy was evaluated using the proposed approach (using $A = |C|$), SSC ($\rho = 1, \alpha = 10$), LRR-H ($\lambda = 0.1$) and LR-SC ($\tau = 0.1$). The parameters of all methods were optimized for best results[8], and for SSC we also projected[9] the column-stacked frames onto their PCA subspace of dimension 9 and segmented the projected frames (excluding this step SSC performance was worse). The results are provided in Table IV, and demonstrate almost perfect segmentation of V_1 (see Fig. 6) using all methods. The segmentation of V_2 was possible only with the proposed approach, while the other methods were unable to segment the 12000 frames due to their complexity.

V. Conclusions

Subspace clustering is a powerful tool for processing and analyzing high dimensional data. This paper presented a low-complexity subspace clustering approach that utilizes sparse representations in conjunction with bipartite graph partitioning. By modeling the relations between the signals according to the atoms that represent them and by exploiting the small number of atoms, the complexity of the proposed approach depends only linearly in the number of signals. Therefore, it is suitable for clustering very large signal collections. Performance evaluation for synthetic data, face clustering and temporal video segmentation demonstrate comparable performance to state-of-the-art at a fraction of the computational load. We further plan to explore the relation between the number of atoms to clustering accuracy, estimation methods for the number of clusters and applications to data corrupted by missing entries and outliers.

[8]SSC was evaluated with the ADMM-based version without outlier support, LRR-H was evaluated with the version that solves equation (3) with post-processing of the affinity matrix, and LR-SC was evaluated with the version that solves Lemma 1 in [9].

[9]The ADMM-based SSC code provides the projection option.

Appendix A
Independent and Disjoint Subspaces

Independent [29] and disjoint subspaces are defined using the sum and the direct sum of a union of subspaces:

Definition 1. *The sum of subspaces $\{\mathbf{S}_i\}_{i=1}^K$ is denoted by $\mathbf{V} = \mathbf{S}_1 + \mathbf{S}_2 + \cdots + \mathbf{S}_K$, such that every $\mathbf{v} \in \mathbf{V}$ equals to $\mathbf{v} = \mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_K$ and $\mathbf{s}_i \in \mathbf{S}_i$.*

Definition 2. *The sum of subspaces $\mathbf{V} = \mathbf{S}_1 + \mathbf{S}_2 + \cdots + \mathbf{S}_K$ is **direct** if every $\mathbf{v} \in \mathbf{V}$ has a **unique** representation $\mathbf{v} = \mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_K$, where $\mathbf{s}_i \in \mathbf{S}_i$. The direct sum is denoted by $\mathbf{V} = \mathbf{S}_1 \oplus \mathbf{S}_2 \oplus \cdots \oplus \mathbf{S}_K$.*

Given the above definitions, we turn now to define independent and disjoint subspaces:

Definition 3. *The subspaces $\{\mathbf{S}_i\}_{i=1}^K$ are independent if their sum is direct. As a consequence, no nonzero vector from any $\mathbf{S}_j$ is a linear combination of vectors from the other subspaces $\mathbf{S}_1, \cdots, \mathbf{S}_{j-1}, \mathbf{S}_{j+1}, \cdots, \mathbf{S}_K$.*

Definition 4. *The subspaces $\{\mathbf{S}_i\}_{i=1}^K$ are disjoint if $\mathbf{S}_i \cap \mathbf{S}_j = \{0\}$ $\forall i \neq j$. Note that independent subspaces are disjoint, however, disjoint subspaces are not necessarily independent.*

Appendix B
Spectral Bipartite Graph Clustering

This appendix provides the derivation of the spectral clustering algorithm for bipartite graphs [22]. Spectral clustering [10] provides an approximate solution to the NP-hard problem of minimizing the normalized-cut criterion. This approach requires the solution of the generalized eigenvalue problem $\mathscr{L}\mathbf{z} = \lambda D\mathbf{z}$, where $\mathscr{L} = D - W$ is the Laplacian and D is diagonal such that $D(i,i) = \sum_{k=1}^{M+L} W(i,k)$. In the bipartite case, the affinity matrix is given by:

$$\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)},$$

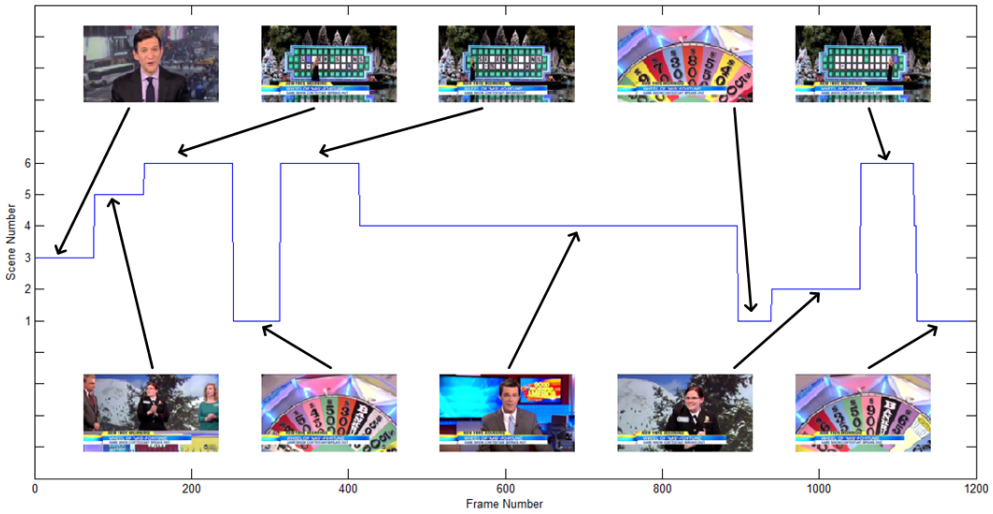


Fig. 6: Temporal video segmentation of V_1 using the proposed approach (98.99% accuracy).

and the Laplacian is given by:

$$\mathscr{L} = \begin{bmatrix} \mathbf{D}_1 & -\mathbf{A} \\ -\mathbf{A}^T & \mathbf{D}_2 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)}, \tag{10}$$

where $\mathbf{D}_1 \in \mathbb{R}^{M\times M}$ and $\mathbf{D}_2 \in \mathbb{R}^{L\times L}$ are diagonal such that

$$\mathbf{D}_1(i,i) = \sum_{j=1}^{L} \mathbf{A}(i,j) \quad \textit{and} \quad \mathbf{D}_2(j,j) = \sum_{i=1}^{M} \mathbf{A}(i,j). \tag{11}$$

The generalized eigenvalue problem can be rewritten as:

$$\begin{bmatrix} \mathbf{D}_1 & -\mathbf{A} \\ -\mathbf{A}^T & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \tag{12}$$

where $\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}$. Equation (12) can be further expanded as follows:

$$\mathbf{D}_1\mathbf{z}_1 - \mathbf{A}\mathbf{z}_2 = \lambda\mathbf{D}_1\mathbf{z}_1 \tag{13}$$

$$-\mathbf{A}^T\mathbf{z}_1 + \mathbf{D}_2\mathbf{z}_2 = \lambda\mathbf{D}_2\mathbf{z}_2. \tag{14}$$

By setting $\mathbf{u} = \mathbf{D}_1^{\frac{1}{2}}\mathbf{z}_1$ and $\mathbf{v} = \mathbf{D}_2^{\frac{1}{2}}\mathbf{z}_2$ the following equations are obtained (assuming non-singularity of $\mathbf{D}_1$ and $Dict2_2$):

$$\mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}\mathbf{v} = (1-\lambda)\mathbf{u} \tag{15}$$

$$\mathbf{D}_2^{-\frac{1}{2}}\mathbf{A}^T\mathbf{D}_1^{-\frac{1}{2}}\mathbf{u} = (1-\lambda)\mathbf{v}, \tag{16}$$

which define the SVD equations of $\overline{\mathbf{A}} = \mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}$:

$$\overline{\mathbf{A}}\mathbf{v}_i = \sigma_i\mathbf{u}_i \text{ and } \overline{\mathbf{A}}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i, \tag{17}$$

where $\mathbf{v}_i$ is the i-th right singular vector, $\mathbf{u}_i$ is the i-th left singular vector and $\sigma_i = 1-\lambda_i$ is the i-th singular value. Therefore, spectral bipartite graph clustering can be obtained from the SVD of $\overline{\mathbf{A}}$, as summarized in algorithm 2, which has a significant complexity advantage over explicit decomposition of the Laplacian, whenever $M \ll L$, since the complexity of the SVD of $\overline{\mathbf{A}}$ is $O(M^2L)$.

Algorithm 2 Spectral Bipartite Graph Clustering

Input: Affinity matrix $\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}$ and number of clusters K.

1) Compute the SVD of $\overline{\mathbf{A}} = \mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}$.
2) Construct the matrix $\mathbf{Z} = \begin{bmatrix} \mathbf{D}_1^{-\frac{1}{2}}\mathbf{U} \\ \mathbf{D}_2^{-\frac{1}{2}}\mathbf{V} \end{bmatrix}$, where $\mathbf{U} = [\mathbf{u}_2 ... \mathbf{u}_K]$ and $\mathbf{V} = [\mathbf{v}_2 ... \mathbf{v}_K]$.
3) Cluster the rows of $\mathbf{Z}$ using the k-means algorithm.

Output: cluster labels for all graph nodes.

Appendix C
Proof of Theorems

The proof of Theorem 1 is composed of two parts: the first part addresses the correctness and uniqueness of the recovery of $\mathbf{C}$ by OMP (as detailed in Algorithm 3), and the second part addresses the correctness of the subspace clustering result by bipartite graph partitioning. The proof relies on the following Lemma:

Lemma 1. *Let $\mathbf{D} \in \mathbb{R}^{N\times M}$ contain K minimal bases for K independent subspaces, then the null-space $\mathcal{N}(\mathbf{D}) = \{0\}$.*

Proof: Let $\{\mathbf{S}_i\}_{i=1}^K$ be a collection of K independent subspaces of dimensions $\{d_i\}_{i=1}^K$, respectively, and let $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N\times d_i}$ is a basis of the i-th subspace and $\sum_i d_i = M \leq N$. Since the subspaces are independent their sum is direct, and every vector $\mathbf{v}$ in their direct sum has a unique representation $\mathbf{v} = \sum_{i=1}^K \mathbf{D}_i\alpha_i$. Equivalently, the solution to the linear system of equations $\mathbf{D}\alpha = \mathbf{v}$ is unique, which leads to $\text{rank}([\mathbf{D}|\mathbf{v}]) = \text{rank}(\mathbf{D}) = M$. Therefore, $\mathbf{D}$ is full rank and $\mathcal{N}(\mathbf{D}) = \{0\}$. ■

Theorem 1. *Let $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_K]$ be a collection of $L = L_1 + L_2 + \cdots + L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^K$. Given a dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N\times d_i}$ spans $\mathbf{S}_i$ and $d_i = dim(\mathbf{S}_i)$, OMP is*

guaranteed to recover the correct and unique sparse representations matrix $\mathbf{C}$ *such that* $\mathbf{Y}=\mathbf{DC}$*, and minimization of the Normalized-Cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.*

Proof: Part I: The matrix $\mathbf{C}$ is computed column-by-column using OMP, therefore, correctness is proved for one column $\mathbf{c}_i=\mathbf{x^k}$ that represents a signal $\mathbf{y}_i\neq 0$ from subspace $\mathbf{S}_i$. OMP terminates either if the residual $\mathbf{r^k}=\mathbf{0}$ or the iteration counter $k=K_{\max}=M$. The proof is provided for each possible termination state of OMP:

i) The residual $\mathbf{r}^k=0$ and the columns of $\mathbf{D}$ selected by the support set Ω^k form exactly $\mathbf{D}_i$ ($\mathbf{S}_i=\text{Span}(\mathbf{D}_{\Omega^k})$): in this case we have $\mathbf{y}_i=\mathbf{Dx^k}=[\mathbf{D_i}\ \mathbf{D_{i^c}}]\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix}$, where $\mathbf{D}_{i^c}$ equals to $\mathbf{D}$ excluding the i-th basis $\mathbf{D}_i$. On the other hand $\mathbf{y}_i$ has a unique representation using $\mathbf{D}_i$ that is given by $\mathbf{y}_i=\mathbf{D}_i\mathbf{c}_*=[\mathbf{D}_i\ \mathbf{D}_{i^c}]\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$. Therefore, we can write $\mathbf{D}\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix}=\mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which can be re-written as: $\mathbf{D}(\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix}-\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix})=0$. Since $\mathcal{N}(\mathbf{D})=\{0\}$, the only solution to this equation is $\mathbf{x_i}=\mathbf{c}_*$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

ii) The residual $\mathbf{r^k}=\mathbf{0}$ and the columns of $\mathbf{D}$ selected by Ω^k include $\mathbf{D}_i$ ($\mathbf{S}_i\subset\text{Span}(\mathbf{D}_{\Omega^k})$): in this case we have $\mathbf{y}_i=\mathbf{Dx^k}=[\mathbf{D_i}\ \mathbf{D_{i^c}}]\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix}$. By using the unique representation of $\mathbf{y}_i$, we obtain $\mathbf{D}\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix}=\mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which can be re-written as: $\mathbf{D}(\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix}-\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix})=0$. Since $\mathcal{N}(\mathbf{D})=\{0\}$, the only solution to this equation is $\mathbf{x_i}=\mathbf{c}_*$ and $\mathbf{x_{i^c}}=\mathbf{0}$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

iii) OMP reached the maximum number of iterations $K_{\max}=M$ and the residual $\mathbf{r^k}\neq\mathbf{0}$: This scenario is impossible as proved in the following. In this case $\mathbf{x}^k$ is the solution of the convex least-squares problem $\arg\min_{\mathbf{x}}\|\mathbf{y}_i-\mathbf{Dx}\|_2$, therefore, the gradient of the least-squares objective equals zero at the global minimum: $\mathbf{D}^T(\mathbf{y}_i-\mathbf{Dx}^k)=0$. By replacing $\mathbf{y}_i$ with its unique representation $\mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$ we obtain $\mathbf{D}^T\mathbf{D}(\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}-\mathbf{x}^k)=0$. Since $\text{rank}(\mathbf{D}^T\mathbf{D})=M$ then $\mathcal{N}(\mathbf{D}^T\mathbf{D})=\{0\}$, and the only solution to this equation is $\mathbf{x^k}=\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which results in $\mathbf{r^k}=\mathbf{0}$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

Part II: Given the correct recovery of $\mathbf{C}$, the collection $\mathbf{Y}$ is decomposed as follows[10]:

[10]This part of the theorem is proved for the case of two subspaces, in order to focus on the essence of the method and avoid cumbersome notations.

Algorithm 3 Orthogonal Matching Pursuit (OMP)

Input: $\mathbf{y}$, $\mathbf{D}=[\mathbf{d_1},\mathbf{d_2},...,\mathbf{d_M}]\in\mathbb{R}^{\mathbf{N}\times\mathbf{M}}$.
Initialize:
1) Iteration counter $k=0$.
2) Maximum number of iterations $K_{\max}=M$.
3) Support set $\Omega^0=\emptyset$.
4) Residual $\mathbf{r^0}=\mathbf{y}$.

Repeat until $\mathbf{r^k}=\mathbf{0}$ **or** $\mathbf{k}=\mathbf{K_{max}}$
1) Increment iteration counter $k=k+1$.
2) Select atom: find $j=\arg\max_j|<\mathbf{r^{k-1}},\mathbf{d_j}>|$.
3) $\Omega^k=\Omega^{k-1}\cup j$.
4) solution $\mathbf{x}^k=\arg\min_{\mathbf{u}}\|\mathbf{y}-\mathbf{Du}\|_2$ s.t. $\text{Support}\{\mathbf{u}\}=\Omega^k$.
5) $\mathbf{r}^k=\mathbf{y}-\mathbf{Dx^k}$

Output: $\mathbf{x^k}$.

$$\mathbf{Y}=[\mathbf{Y}_1\ \mathbf{Y}_2]=\mathbf{DC}=[\mathbf{D}_1\ \mathbf{D}_2]\begin{bmatrix}\mathbf{C}_1&0\\0&\mathbf{C}_2\end{bmatrix}. \tag{18}$$

By defining $\mathbf{A}_1=|\mathbf{C}_1|\in\mathbb{R}^{d_1\times L_1}$ and $\mathbf{A}_2=|\mathbf{C}_2|\in\mathbb{R}^{d_2\times L_2}$, the affinity matrix is given by:

$$\mathbf{W}=\left[\begin{array}{cc|cc}\multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}}&\mathbf{A}_1&0\\&&0&\mathbf{A}_2\\\hline\mathbf{A}_1^T&0&\multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}}\\0&\mathbf{A}_2^T&&\end{array}\right].$$

The optimal partition is $\mathcal{V}_1=\{d_1\ atoms\ of\ \mathbf{D}_1\cup L_1\ signals\ spanned\ by\ \mathbf{D}_1\}$ and $\mathcal{V}_2=\{d_2\ atoms\ of\ \mathbf{D}_2\cup L_2\ signals\ spanned\ by\ \mathbf{D}_2\}$. W.l.o.g. we rearrange the rows and columns of W such that the vertices associated with $\mathcal{V}_1$ are the leading vertices and the vertices associated with $\mathcal{V}_2$ are the tailing vertices. The rearranged affinity is given by:

$$\overline{\mathbf{W}}=\left[\begin{array}{cc|cc}0&\mathbf{A}_1&\multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}}\\\mathbf{A}_1^T&0&&\\\hline\multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}}&0&\mathbf{A}_2\\&&\mathbf{A}_2^T&0\end{array}\right].$$

The cut of the optimal partition is given by:

$$\text{cut}(\mathcal{V}_1,\mathcal{V}_2)=\sum_{i\in\mathcal{V}_1,j\in\mathcal{V}_2}\overline{\mathbf{W}}_{ij}=0, \tag{19}$$

and the weight of each group is given by:

$$\text{weight}(\mathcal{V}_{1,2})=\textstyle\sum_{i\in\mathcal{V}_{1,2}}\sum_k\overline{W}_{ik}=2S(\mathbf{A}_{1,2})>0, \tag{20}$$

where $S(\mathbf{Q})=\sum_{n,m}\mathbf{Q}_{nm}$ is the sum of matrix entries. Therefore, the normalized-cut metric equals zero for the optimal partition.

■

Theorem 2. *Let* $\mathbf{Y}=[\mathbf{Y}_1,\mathbf{Y}_2,\cdots,\mathbf{Y}_K]$ *be a collection of* $L=L_1+L_2+\cdots+L_K$ *signals from K independent subspaces of dimensions* $\{d_i\}_{i=1}^K$*. Given a dictionary* $\mathbf{D}=[\mathbf{D}_1,\mathbf{D}_2,\cdots,\mathbf{D}_K]$ *such that* $\mathbf{D}_i\in\mathbb{R}^{N\times d_i}$ *spans* $\mathbf{S}_i$ *and* $d_i>dim(\mathbf{S}_i)$*, OMP is guaranteed to recover a correct sparse representations matrix* $\mathbf{C}$ *such that* $\mathbf{Y}=\mathbf{DC}$*,* $\mathbf{C}$ *include* only *atoms from the correct subspace basis for each signal, and minimization of the*

Normalized-Cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.

Proof: The matrix $\mathbf{C}$ is computed column-by-column using OMP, therefore, correctness is proved for one column $\mathbf{c}_i = \mathbf{x^k}$ that represents a signal $\mathbf{y}_i \neq 0$ from subspace $\mathbf{S}_i$. OMP terminates either if the residual $\mathbf{r^k} = \mathbf{0}$ or the iteration counter $k = K_{\max} = M$. The proof is provided for each possible termination state of OMP:

i) $\mathbf{r^k} = \mathbf{0}$ and $\mathbf{S}_i = \text{Span}(\mathbf{D}_{\Omega^k})$: in this case we have $\mathbf{y}_i = \mathbf{D}\mathbf{x^k} = [\mathbf{D_i}\ \mathbf{D_{i^c}}] \begin{bmatrix} \mathbf{x}_i \\ 0 \end{bmatrix} = \mathbf{D_i}\mathbf{x_i}$, and $\mathbf{x}_i \neq 0$. Therefore, $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.

ii) $\mathbf{r^k} = \mathbf{0}$ and $\mathbf{S}_i \subset \text{Span}(\mathbf{D}_{\Omega^k})$: in this case we have $\mathbf{y}_i = \mathbf{D}\mathbf{x^k} = [\mathbf{D_i}\ \mathbf{D_{i^c}}] \begin{bmatrix} \mathbf{x_i} \\ \mathbf{x_{i^c}} \end{bmatrix}$. On the other hand, $\mathbf{x^k}$ is the solution to the least-squares problem 4) of Algorithm 3, which is computed using the pseudo-inverse $\mathbf{x}^k = \mathbf{D}^{\dagger}_{\Omega^k}\mathbf{y}_i$. Therefore, this solution is guaranteed to have the smallest l_2-norm among all feasible solutions to the equation $\mathbf{y}_i = \mathbf{D}\mathbf{u}$ (s.t. support($\mathbf{u}$)=Ω^k). Since $\mathbf{y}_i \in \mathbf{S}_i$ it can be represented by $\mathbf{y}_i = \mathbf{D}_i\mathbf{c}_* = [\mathbf{D}_i\ \mathbf{D}_{i^c}] \begin{bmatrix} \mathbf{c}_* \\ 0 \end{bmatrix}$, which leads to $\mathbf{D}_i\mathbf{c}_* = \mathbf{D_i}\mathbf{x_i} + \mathbf{D_{i^c}}\mathbf{x_{i^c}}$. Note that this equation can be rewritten as[11] $\mathbf{D}_i(\mathbf{c}_* - \mathbf{x_i}) = \mathbf{D_{i^c}}\mathbf{x_{i^c}}$, in which the left-hand side is a vector in $\mathbf{S}_i$ and the right-hand side is a vector in $\oplus_{j=1,j\neq i}^{K}\mathbf{S}_j$. The subspaces $\mathbf{S}_i$ and $\oplus_{j=1,j\neq i}^{K}\mathbf{S}_j$ are independent, therefore their intersection contains only the null vector. The implications of this result are that $\mathbf{D}_{i^c}\mathbf{x_{i^c}} = \mathbf{0}$ and that $\mathbf{x_i}$ is a feasible solution (namely $\mathbf{y}_i = \mathbf{D}_i\mathbf{x_i}$). Since the pseudo inverse-based solution provides the solution with the smallest l_2-norm, we obtain that $\left\| \begin{bmatrix} \mathbf{x_i} \\ 0 \end{bmatrix} \right\|_2 < \left\| \begin{bmatrix} \mathbf{x_i} \\ \mathbf{x_{i^c}} \end{bmatrix} \right\|_2 \ \forall\ \mathbf{x_{i^c}} \neq \mathbf{0}$). Therefore, this solution must lead to $\mathbf{x_{i^c}} = \mathbf{0}$ and thus $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.

iii) OMP reached the maximum number of iterations $K_{\max} = M$: In this case there is an infinite number of solutions to the equation $\mathbf{y}_i = \mathbf{D}_{\Omega^M}\mathbf{x}^k = \mathbf{D}\mathbf{x}^k = \mathbf{D}_i\mathbf{x_i} + \mathbf{D_{i^c}}\mathbf{x_{i^c}}$, such that $\mathbf{D_{i^c}}\mathbf{x_{i^c}} = \mathbf{0}$. Therefore, the minimizer of the convex least-squares problem $\arg\min_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_2$ must reach its global minimum, which is $\mathbf{r^k} = \mathbf{0}$, and following case ii) above, $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.

The second part of the theorem follows exactly from part II of therom I. ■

REFERENCES

[1] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

[2] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination condtions. *CVPR*, 2003.

[3] Y. M. Lu and M. N. Do. A theory for sampling signals from a union of subspaces. *IEEE Trans. on Signal Processing*, 56(6), 2008.

[4] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2), 2011.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering. *CVPR*, 2009.

[6] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11), 2013.

[7] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. *ICML*, 2010.

[8] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 98(1), 2013.

[9] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution for robust subspace estimation and clustering. *CVPR*, 2011.

[10] U. V. Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4), 2007.

[11] S. Rao, R. Tron, Y. Ma, and R. Vidal. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. *CVPR*, 2008.

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartgraphy. *Commun. ACM*, 1981.

[13] J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenarate. *ECCV*, 2006.

[14] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 1998.

[15] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervized clustering. *ICASSP*, 2010.

[16] A. Adler, M. Elad, and Y. Hel-Or. Probabilistic subspace clustering via sparse representations. *IEEE Signal Processing Letters*, 20(1), 2013.

[17] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image Processing*, 15(12), 2006.

[18] M.D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M.E. Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995 –1005, june 2010.

[19] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Asilomar Conf. Signals, Systems, and Computers*, 1993.

[20] R. Rubinstein, A.M. Bruckstein, and M.Elad. Dictionaries for sparse representation modeling. *Proc. of the IEEE*, 98(6), 2010.

[21] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11), 2006.

[22] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *SIGKDD*, 2001.

[23] L. N. Trefethen and D. Bau III. *Numerial Linear Algebra*. SIAM, 1997.

[24] M.A. Davenport and M.B. Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. *Information Theory, IEEE Transactions on*, 56(9), 2010.

[25] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.

[26] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. *CVPR*, 1991.

[27] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 25(2), 2003.

[28] A.S Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[29] K. Hoffman and R. Kunze. *Linear Algebra*. Prentice-Hall, 1971.

[11] The following argument relies on Theorem 1 in [6].