Sparse Subspace Representation for Spectral Document Clustering

Saha Budhaditya and Dinh Phung and Duc-Son Pham and Svetha Venkatesh

***Abstract*—We present a novel method for document clustering using sparse representation of documents in conjunction with spectral clustering. An l_1-norm optimization formulation is posed to learn the sparse representation of each document, allowing us to characterize the affinity between documents by considering the overall information instead of traditional pairwise similarities. This document affinity is encoded through a graph on which spectral clustering is performed. The decomposition into multiple subspaces allows documents to be part of a sub-group that shares a smaller set of similar vocabulary, thus allowing for cleaner clusters. Extensive experimental evaluations on three real-world datasets from TDT2, Reuters-21578 and 20Newsgroup corpora show that our proposed method consistently outperforms state-of-the-art algorithms. Significantly, the performance improvement over other methods is prominent in more complex datasets.**

***Keywords*-sparse representation, document clustering**

I. Introduction

Document clustering, a long standing problem, aims to group similar documents to facilitate higher level tasks - document organization, indexing and search. A common approach is to use the bag-of-word representation wherein each document is a vector of size V, the number of distinct words in the corpus. This representation creates a term-by-document matrix that lends itself to systematic analysis - for example, singular value decomposition can be performed on this matrix, to project the document vector into lower dimensional space. Subsequent steps such as clustering can then be performed. Despite its popularity and success, this class of methods use a single subspace to represent the data, potentially leading to sub-optimal representations and inferior performance in challenging datasets.

We propose a novel framework for document clustering, using multiple subspace representation of documents to construct a sparse graph on which spectral clustering can be performed. Sparse representation approach, popular in signal processing [2], [14], [21], [25],[28], is relatively new for document representation. Sparse subspace clustering (SSC) constructs a sparse representation of the data, and its key advantage is its ability to automatically discover the number of subspaces as well as their dimensions. In addition, SSC employs ℓ_1-norm regularization solvers, typically polynomial in complexity, allowing it to scale well with data.

Note: This is the author version of the published conference paper in ICDM 2012.

Budhaditya, Phung, and Venkatesh are with Deakin University. Pham is with Curtin University.

Leveraging this formulation to text document modeling, using the bag-of-word representation, we seek to represent each document vector $\mathbf{x}_i$ as a linear combination of other documents in the corpus $\mathbf{x}_i = \sum_{j\neq i} c_{ij}\mathbf{x}_j = \sum_{j\in S_i, j\neq i} c_{ij}\mathbf{x}_j + \sum_{j\notin S_i} c_{ij}\mathbf{x}_j$ where c_{ji} (s) are the coefficients, and S_i denotes index set (subspace) of documents that document $\mathbf{x}_i$ belongs to. In the ideal case, the coefficients in the second term are zeros, giving rise to a sparse representation. In addition, under the bag-of-word representation if two documents $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same subset of distinct vocabulary, then c_{ij} is expected to be non-zero, otherwise it is zero. This process will essentially induce different subspaces for different subsets of documents, each of which possess distinct *smaller* subsets of vocabulary. This is expected to produce finer grain, reduced dimension representations as compared with traditional methods such as LSI, that consider a single subspace, thus improving cluster quality.

The representation is learned by minimizing the objective function for reconstruction, using the linear combination of documents and a ℓ_1-norm based optimization method. Further, we extend it for an affine combination of documents and in noisy settings. We then construct a sparse graph whose vertices correspond to documents and the edge weights are determined from our l_1-norm optimization solution. Using this graph, a spectral clustering algorithm is then performed to cluster documents . The advantage of this method is two-fold: we learn the neighborhood and affinity scores between given data points and neighbors by considering the overall information in document space instead of *pairwise* similarities using euclidean distance. Property of ℓ_1-norm based optimization methods makes the resultant graph naturally sparse.

We evaluate our proposed method extensively using three real-world datasets and several state-of-the-art document clustering algorithms. Datasets include: Topic detection and tracking corpus (TDT2), and from news corpus: Reuters-21578, and 20-Newsgroup. Methods to be compared against are recently proposed document clustering algorithms, including locality semantic indexing (LSI) [8], locality preserving indexing (LPI) [16], graph regularized NMF (GNMF) [5], *symmetric NMF* [17], Laplacian embedding (LE) [3], and locally consistent concept factorization (LCCF) [4]. We also evaluate against k-NN or ϵ-ball based methods [17]. In addition, we compare our results against semi-supervised algorithm in [23] where supervision is provided in terms of similarity constraints between the docu-

ments part-based representation. We show that our proposed method consistently outperforms all these methods. We show that affine and noisy sparse representations yield even better results.

Addressing the problem of document clustering, our contributions are:

- A novel formulation of document clustering using *using sparse representations*, enabling multiple sub-space decomposition. This enables clustering via the derived multiple subspaces;
- Extensive validation of the proposed method on three popular real world data sets, compared against state-of-art benchmark spectral and non-negative matrix factorization algorithms. The resultant improvement is significant - on the challenging 20 Newsgroup dataset, we outperform other methods, on Rand-index by 11-32%, and on F-measure by 16-40%

The novelty of our work is that it does not require parameter tuning, for example neighborhood size or distance measures, to learn the affinity matrix, crucial for other spectral clustering algorithms. The constructed graph is *naturally* sparse. The significance of $l_1$norm based spectral methods for document clustering is that it offers a systematic approach to learn the neighborhood structure of data points, followed by recovery of subspace for each cluster in the corpus. This not only facilitates representation of clusters, but enables capture of underlying semantic concepts in sub-groups at finer levels.

II. Related Background

Recently, the literature over different disciplines has revealed the usefulness of sparse representation, which is related to the theory of compressed sensing [7], [13]. Successful applications include robust face recognition [26], motion segmentation [14], image coding [2], image restoration [18], and image super-resolution [27]. It is the success of these methods in related fields that motivates us to explore sparse representations in the document clustering context. We note importantly that the sparse representation here is made with respect to other observed data and it is not the sparse nature of documents with respect to a large and fixed vocabulary.

There are two main approaches in the document clustering literature, namely matrix factorization and spectral clustering. Examples of matrix factorization clustering methods include graph regularized non-negative matrix factorization (NMF) [5], [17], symmetric NMF[17], locally consistent concept factorization (LCCF) [4]. The NMF-variant methods usually decompose data matrix into two nonnegative matrices consisting of basis and coefficient vectors respectively. They apply k-means on the coefficient matrix to group similar documents. The major limitation of NMF methods is that they only learn the global structure of the document space and ignore the local structure between documents. Besides, NMF methods can be computationally expensive.

The second approach which we follow in this work is spectral clustering. Examples of this category include latent semantic indexing (LSI) [29], locality preserving indexing (LPI) [16], co-clustering and its variants [9], graph-cut methods [12][30] such as normalized cut [10], [22], ratio cut [11], min-max cut [12], normalized spectral clustering using Ng, Jordan and Weiss (NJW) method. Spectral methods treat a corpus as a graph whose vertices represent documents. The edges of the graph encode the notion of similarity between documents and is typically summarized by the affinity matrix. To perform clustering, a spectral method computes the graph Laplacian $\mathbf{L}$, which is a function of the affinity matrix $\mathbf{S}$. For effective clustering, spectral methods seek a transformation on the original high-dimensional document data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ to a lower-dimensional space $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_N]$ via a linear transformation matrix $\mathbf{P}$, so that $\mathbf{y}_i = \mathbf{P}^T\mathbf{x}_i$. This transformation matrix is typically found from an eigenvalue problem. Define $\mathbf{D}$ a diagonal matrix, whose the ith diagonal entry being $D_{ii} = \sum_{j=1}^{n} S_{ij}$. Different spectral methods can be distinguished by different choice of the graph Laplacian, the eigenvalue problem, and the final partition of the projected data. For example, locality preserving indexing (LPI) [16] uses $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and computes $\mathbf{P}$ as the matrix of eigenvectors associated with the smallest eigenvalues of the problem $\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{p} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{p}$; latent semantic indexing (LSI) [8] uses $\mathbf{L} = \mathbf{I}$ and computes $\mathbf{P}$ as the matrix of eigenvectors associated with the largest eigenvalues of the problem $\mathbf{X}\mathbf{X}^T\mathbf{p} = \lambda\mathbf{p}$; Laplacian embedding (LE) [3] computes $\mathbf{P}$ as the matrix of eigenvectors associated with the smallest eigenvalues of the problem $\mathbf{L}\mathbf{p} = \lambda\mathbf{D}\mathbf{p}$. It is noted that LPI, LSI, and LE all use k-means on the projected data for final clustering. On contrary to the above methods, graphcut variants perform the final clustering of the data points slightly differently. For example, normalized spectral clustering using Shi-Malik's method (SM) [22] computes $\mathbf{P}$ as the matrix of eigenvectors associated with the smallest eigenvalues of the problem $\mathbf{L}\mathbf{p} = \lambda\mathbf{D}\mathbf{p}$. The cluster assignment is obtained by using k-means on the rows of the matrix $\mathbf{P}$. Similarly, normalized spectral clustering using Ng-Jordan-Weiss method (NJW) [20] chooses $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$ and computes $\mathbf{P}$ as the matrix of eigenvectors associated with the smallest eigenvalues of $\mathbf{L}$, where each row of $\mathbf{P}$ is normalized to unity, and k-means is used on the rows of $\mathbf{P}$ for final clustering. Ratio Cut [24] finds the second largest eigenvalue of $(\mathbf{D} - \mathbf{S})\mathbf{p} = \lambda\mathbf{p}$. For clustering, the entries of the eigenvector corresponding to second largest eigenvalue, i.e. $\mathbf{p}_2$ is sorted in increasing order and the order list is used to partition the data into two parts in a manner so that the cut criterion is minimized over the two partition. This process continues recursively until the number of desired clusters is obtained. Min-Max Cut [12], [30] also solves the eigenvalue problem $\mathbf{S}\mathbf{p} = \lambda\mathbf{D}\mathbf{p}$, and then use second largest eigenvalue

$\mathbf{p}_2$ to partition the data points.

Regardless of the choices for $\mathbf{L}$, all spectral methods depend on the input affinity matrix $\mathbf{S}$. We argue that a well-designed affinity matrix that characterizes the underlying statistics of documents is the key to success. We note previous works typically use either heuristic or non-scalable choices for the affinity matrix. For example, [4], [16], [17] use k-nearest neighbour (NN); [24],[22] use ϵ-NN for computing the affinity matrix, which necessitates heuristic tuning for k or ϵ. Other methods, such as [20],[12], [30], use fully-connected graphs, which are not suitable for large-scale document clustering. On the contrary, our method is scale invariant, i.e. it computes $\mathbf{S}$ automatically regardless of varying data scales. The other important aspect is that our method discovers multiple subspaces having small vocabulary sets which represent unique categories in the corpus. This is not possible with other methods that could only learn a single subspace.

III. Proposed Document Clustering Framework

Our proposed method consists of two stages. In the first stage, we obtain sparse representations for documents using either linear subspace, affine subspace, or noisy formulations. In the second stage, we construct the affinity matrix from the sparse representations and use a version of normalized spectra clustering with NJW. We detail each stage as follows.

A. Sparse Subspace Representations

Consider a set of N documents represented by a term-by-document data matrix $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_N]$, where $\tilde{\mathbf{x}}_i \in \mathbb{R}^D$ and D is the number of distinct vocabularies. We first normalize the data by performing an SVD $\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and project each document vector $\tilde{\mathbf{x}}_i$ into a lower r-dimensional space: $\mathbf{x}_i = \mathbf{U}_r^T \tilde{\mathbf{x}}_i$ and transform the original data matrix $\tilde{\mathbf{X}}$ into $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

Our proposed method then seeks a sparse representation for data points in $\mathbf{X}$. Our goal is represent each document $\mathbf{x}_i$ as a linear combination other documents. Intuitively, only document $\mathbf{x}_j$ which is closely related to $\mathbf{x}_i$ will contribute to the construction of $\mathbf{x}_i$ and vice versa. For instance, in the extreme example of a document collection, which consists of only two non-overlapped sub-categories of documents, our approach of representation is expected to induce exactly two subspaces. In practice, there will be some overlapping between subcategories and our formulation shall quantify them exactly through the linear combination coefficients computed from an optimization problem.

For each document vector $\mathbf{x}_i$, denote by $\mathcal{S}_i$ the index set of the subspace (sub-collection of documents) that $\mathbf{x}_i$ belongs to, we rewrite the linear representation for $\mathbf{x}_i$ as follows:

$$\mathbf{x}_i = \sum_{j \neq i} c_{ij} \mathbf{x}_i = \sum_{i \in \mathcal{S}_i, j \neq i} c_{ij} \mathbf{x}_i + \sum_{j \notin \mathcal{S}_i} c_{ij} \mathbf{x}_i. \quad (1)$$

Ideally, when there is no overlapping between the subspaces, the coefficients in the second summation of the right term in Equation (1) are zeros, giving rise to a sparse representation. By collecting the linear representation of all points $\mathbf{x}_i$ in the coefficient matrix $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N]$, one can express the representation in the matrix form as follows:

$$\mathbf{X} = \mathbf{X}\mathbf{C}, \ \text{diag}(\mathbf{C}) = 0.$$

Define the ℓ_1-norm of a matrix $\mathbf{C}$ as $||\mathbf{C}||_1 = \sum_{i,j} |c_{ij}|$. It follows from compressed sensing that minimizing this ℓ_1-norm naturally promotes sparsity. The above equality constraints ensure the solution is consistent with the observed data. Here, there are several choices that one may need to impose to recover the sparse coefficients as follows:

1) Linear subspace formulation: Under this formulation, there are no further constraints on $\mathbf{C}$ and the sparse represent ions of documents are obtained by solving following optimization problem:

$$\arg\min_{\mathbf{C}} ||\mathbf{C}||_1 \quad (2)$$

$$\text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{C}, \ \text{diag}(\mathbf{C}) = \mathbf{0}. \quad (3)$$

Ideally, equation 3 is related to the sparse subspace representation of the data points in $\mathbf{X}$ which will be explored in details in later part of this section. This optimization problem is convex with equality constraints, and hence it is readily solved with existing convex optimization packages. In compressed sensing, this is referred to as the basis pursuit problem.

2) Affine subspace formulation: The linear subspace formulation does not constrain the search space for the coefficient matrix. In certain scenarios, it is observed that constraining the parameters by an affine constraints narrows the search space, and hence may improve numerical stability and enhance subspace separation. To impose affine constraints to the case here, we can represent each document $\mathbf{x}_i$ as an affine combination of other documents as follows:

$$\mathbf{x}_i = \textstyle\sum_{j \neq i} c_{ij} \mathbf{x}_j \quad (4)$$

$$\text{s.t.} \quad \textstyle\sum_{j=1}^{N} c_{ij} = 1 \quad (5)$$

Then the sparse representation of documents are found from the following problem:

$$\arg\min_{\mathbf{c}_i} ||\mathbf{c}_i||_1 \quad (6)$$

$$\text{s.t.} \quad \mathbf{x}_i = \mathbf{X}\mathbf{c}, \ \mathbf{c}_i^T \mathbf{1} = 1, \ c_{ii} = 0 \quad . \quad (7)$$

We show that the machinery for solving the linear subspace formulation can be readily used here. To simplify the notation, denote $\mathbf{X}^{-i}$ as $\mathbf{X}$ with the i^{th} column removed, and $\mathbf{a}$ as $\mathbf{c}_i$ with the i^{th} entry removed. We can rewrite the formulation in the following form

$$\arg\min_{\mathbf{a}} ||\mathbf{a}||_1 \quad (8)$$

$$\mathbf{x}_i^{'} = \mathbf{X}^{'}\mathbf{a}. \quad (9)$$

Here, $\mathbf{x}_i^{'}$= $\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$ and $\mathbf{X}^{'}$ = $\begin{bmatrix} \mathbf{X}^{-i} \\ \mathbf{1^T} \end{bmatrix}$. Thus, the affine formulation has the same form as the linear formulation and thus efficient sparsity solvers can be readily used.

3) Noisy data formulation: In practice, it is more appropriate to account for noise when modeling documents as being sampled from the subspaces. In such cases, we can express each document as $\mathbf{x}_i = \bar{\mathbf{x}}_i + \mathbf{e}_i$, where $\bar{\mathbf{x}}_i$ is the true representation of the i^{th} document and $\mathbf{e}_i$ is the noise, which is bounded as $||\mathbf{e}_i||_2 \leq \varepsilon$. Extending the affine subspace formulation to account for noise, we propose to compute the sparse representation $\mathbf{x}_i$ by solving the following optimization problem:

$$\arg\min_{\mathbf{C}} ||\mathbf{c}_i||_1 \quad (10)$$

$$\text{s.t.} \quad ||\mathbf{x}_i - \mathbf{X}\mathbf{c}_i||_2^2 \leq \varepsilon, \quad \sum_i c_{ij} = 1,\ c_{ii} = 0. \quad (11)$$

We next transform the formulation to a familiar form that can be efficiently solved using existing convex optimization solvers. To simplify the notation, denote $\mathbf{X}^{-i}$ as $\mathbf{X}$ with the ith column removed, and $\mathbf{a}$ as $\mathbf{c}_i$ with the i^{th} entry removed. The above formulation can be rewritten as

$$\underset{\mathbf{a}}{\text{argmin}} \quad ||\mathbf{a}||_1 \quad (12)$$

$$\text{s.t.} \quad \|\mathbf{x}_i - \mathbf{X}^{-i}\mathbf{a}\|_2^2 \leq \varepsilon \quad (13)$$

$$\textstyle\sum_i a_i = 1. \quad (14)$$

To solve this problem, we find it more convenient to express in the Lagrangian form, and our goal is to minimize the following objective function

$$\mathcal{L}(\mathbf{a}, y) = ||\mathbf{a}||_1 + \frac{\mu}{2}||\mathbf{x}_i - \mathbf{X}^{-i}\mathbf{a}||_2^2 + y(1 - \mathbf{1^T}\mathbf{a}) + \frac{\eta}{2}(1 - \mathbf{1^T}\mathbf{a})^2, \quad (15)$$

with respect to $\mathbf{a}$ and the Lagrangian variable y. Here, μ is the Lagrangian equivalence for the noise bound ε, and η is a parameter to improve numerical stability, which can be set to a small number. Following the alternative Lagrangian multiplier framework [6], we can solve this problem by alternating between y and $\mathbf{a}$ in an iterative fashion. The complete derivation of minimizing $\mathcal{L}(\mathbf{a}, \mathbf{y})$ is detailed in the Appendix.

B. Spectral Document Clustering

1) Affinity Graph Construction: After obtaining the coefficient matrix $\mathbf{C}$, we have the sparse representation of each document as $\mathbf{c}_i$ where the nonzero coefficients of $\mathbf{c}_i$ correspond to documents from the same subspace. The next step is to group the documents into multiple linear subspaces where each subspace corresponds to similar documents. Under the spectral approach, an undirected graph $G_{\mathbf{C}}$ is constructed on $\mathbf{X}$, where each vertex of $G_{\mathbf{C}}$ is a document. The affinity matrix $\mathbf{S}$ is constructed as $\mathbf{S_C} = |\mathbf{C}| + |\mathbf{C}|^T$. Specifically in our case, the connected components of $G_{\mathbf{C}}$ correspond to the nonzero coefficients of $\mathbf{C}$. Thus, documents corresponding to the same subspace are connected, whilst documents belonging to the different subspace are not connected. If the data is sorted according to their similarities and if there are K connected components in graph $G_{\mathbf{C}}$, then $G_{\mathbf{C}}$ will have a block-diagonal matrix as follows

$$\mathbf{S_C} = \begin{bmatrix} \mathbf{S_{1C}} & 0 & . & . & 0 \\ 0 & \mathbf{S_{2C}} & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & \mathbf{S}_{K\mathbf{C}} \end{bmatrix},$$

where $\mathbf{S}_{k\mathbf{C}}$ is the affinity matrix is of data points in subspace $\mathcal{S}_k$. In Figures 1 and 2, we give examples of these affinity matrices obtained from three datasets used in our experiments.

2) Clustering: Recall that $\mathbf{P}$ is a transformation matrix, which maps data points in $\mathbf{X}$ onto a lower dimensional space $\mathbf{Y}$, where $\mathbf{y}_i = \mathbf{P}^T\mathbf{x}_i$. In our proposed framework, we compute $\mathbf{P}$ as the matrix of principal eigenvectors of the NJW Laplacian matrix, which is defined as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S_C}\mathbf{D}^{-\frac{1}{2}}.$$

Here, $\mathbf{D}$ is a diagonal matrix with diagonal entries being $D_{ii} = \sum_j \mathbf{S}_{\mathbf{C},ij}$ where $\mathbf{S}_{\mathbf{C},ij}$ is the entries at the position (i, j) of $\mathbf{S_C}$. To obtain K clusters, we select $K+1$ principal eigenvectors to construct $\mathbf{P}$, i.e. $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{K+1}]$. This is a $N \times (K+1)$ matrix , where the i-th document is represented by the coefficients of the i-th row of $\mathbf{P}$. We normalize each row of $\mathbf{P}$ to unity and use k-means on these rows. If K is unknown, it can be found by counting the number of smallest eigenvalues of the Laplacian matrix that are or close to zero.

The overall method for the linear subspace formulation is summarized in Algorithm 1 (see next page).

IV. Experiments

Three real-world benchmark datasets are used: Topic detection and tracking (TDT2), Reuters-21578 and 20-Newsgroup corpus. TDT2[1] was collected from 6 sources, including 2 news wires (APW and NYT), two radio programs (VOA and PRI) and two TV channel (CNN and ABC). Reuters-21578[2] contains 21578 documents across 135 cat-

[1]Available at: http://www.nist.gov/speech/tests/tdt/tdt98/index.html and detailed description as well as preprocessed data can be found in [1].

[2]http://www.davidlewis.com/resources/testcollections/reuters21578/.

Algorithm 1 Document Clustering via Sparse Spectral Graph Partitioning (SSGP)

Input: Documents in lower subspace $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$.
Output: K clusters of $\mathbf{X}$

- Compute the sparse coefficients matrix $\mathbf{C}$ solving the optimization problem:

$$\arg\min_{\mathbf{C}} ||\mathbf{C}||_1 \quad s.t.\ \mathbf{X} = \mathbf{X}\mathbf{C}$$

- Compute affinity matrix $\mathbf{S_C} = |\mathbf{C}| + |\mathbf{C}|^T$.
- Computing the NJW Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S_C}\mathbf{D}^{-\frac{1}{2}}$.
- Perform eigenvalue decomposition of $\mathbf{L}$.
- Compute $\mathbf{P}$ as the matrix of $K+1$ eigenvectors associated with the smallest eigenvalues.
- Each row of $\mathbf{P}$ is normalized to unity.
- The clusters are obtained by applying k-means on the rows of the normalized $\mathbf{P}$.

egories. 20Newsgroup[3] contains approximately 20,000 documents divided into 20 groups or categories. There is great overlap among the groups (e.g., *comp.graphics*, *ibm.pc.hardware*, *mac.pc.hardware*, *comp.windows*), and this makes 20Newsgroup a very challenging dataset for clustering.

In all cases, we remove duplicate documents across categories and retain only categories containing no less than 10 documents. We also perform a standard pre-preprocessing step, including removal of stop words and stemming. Each corpus is then represented by a term-document matrix, where rows correspond to the vocabulary in the corpus and columns correspond to the documents. The final statistics used in our experiment for each of these datasets is summarized in Table I.

Table I
STATISTICS FOR DATASETS USED

Datasets	Size (N)	Vocab size	# Categories
TDT2	10021	36771	30
Reuters-21578	8213	18933	25
20Newsgroup	20,000	16437	20

A. Evaluation Metrics

Clustering results are evaluated by comparing the true labels of the documents against labels obtained from the algorithms. We used standard evaluation metrics [4], [11], [15], including: *clustering accuracy* (AC), *normalized mutual information* (NMI), *rand index* (RI) and *F-measure* F_β. We briefly describe them below.

[3] http://www.people.csail.mit.edu/jrennie/20Newgroups/

Clustering Accuracy (AC) : AC quantifies the accuracy of trying to map one-to-one between true class label and obtained class label as described in [19]. Given document $\mathbf{x}_i$, if t_i and o_i are true class and obtained class labels, then

$$AC = \frac{\sum_{i=1}^{N} \delta(t_i, \text{map}(o_i))}{N},$$

where $\delta(p,q) = 1$ only if $p = q$ or 0 otherwise. The map(o_i) is permutation mapping function that try to map obtained label o_i to the most suitable true class label (see [19] for further details).

Normalized Mutual Information (NMI): NMI quantifies the quality of obtained clusters with respect to true clusters. If T denotes the true clustering result and O the obtained clustering results, the mutual information is first defined as:

$$\text{MI}(T,O) = \sum_{\mathbf{t}_i \in T} \sum_{\mathbf{o}_j \in O} p(\mathbf{t}_i, \mathbf{o}_j) \log \frac{p(\mathbf{t}_i, \mathbf{o}_j)}{p(\mathbf{t}_i)p(\mathbf{o}_j)},$$

where $p(t_i) = \frac{|\mathbf{t}_i|}{N}$, $p(\mathbf{o}_j) = \frac{|\mathbf{o}_j|}{N}$, $p(\mathbf{t}_i, \mathbf{o}_j) = \frac{|\mathbf{t}_i \cap \mathbf{o}_j|}{N}$, and $|\mathbf{t}_i|$ denotes the number of data points in cluster $\mathbf{t}_i$ and $|\mathbf{t}_i \cap \mathbf{o}_j|$ is the the number of data points belong to both clusters $\mathbf{t}_i$ and $\mathbf{o}_j$. Normalized Mutual Information between $T$and O is then defined as:

$$\text{NMI}(T,O) = \frac{\text{MI}(T,O)}{\max(H(T), H(O))},$$

where $H(T)$ and $H(O)$ are the entropies for T and O respectively. NMI ranges from 0 and 1 and $\text{NMI}(T,O) = 0$ implies T and O are disjoint whereas $\text{NMI}(T,O) = 1$ implies T and O are identical or a perfect clustering result has obtained.

Rand Index (RI): If a true positive (TP) is scored when two similar documents in the groundtruth are grouped together in the obtained results, a true negative (TN) is when two dissimilar documents are grouped separately, a false positive (FP) is when two dissimilar documents are grouped together and a false negative (FN) is when two similar documents are grouped separately, then the *rand index* (RI) is defined as follows:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}.$$

Precision (P) , *Recall* (R) and *F-measure* (F_β): are also defined as:

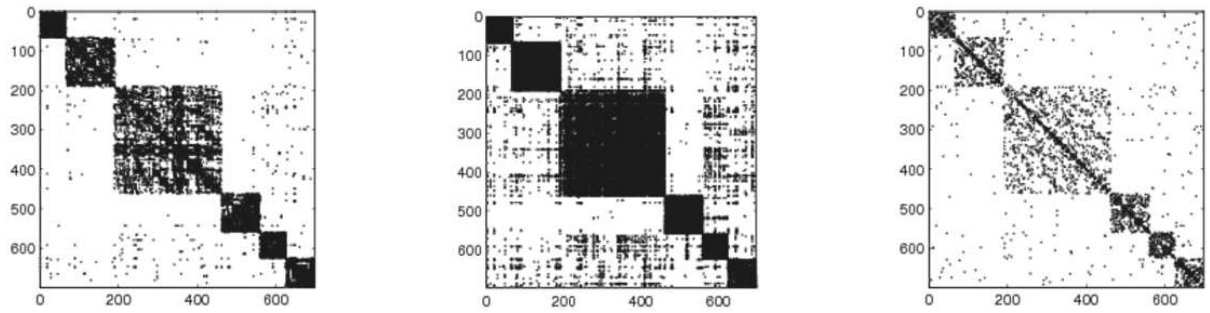
$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN},$$
$$F_\beta = \frac{(\beta^2 + 1)P \times R}{P + R}$$

B. Results and Comparison

We extensively compare our proposed clustering framework (denoted by SSGP) against recently proposed state-of-art document clustering algorithms from two main approaches: spectral and nonnegative matrix factorization.

Table II
PERFORMANCE ON TDT2 DATA SETS

K	Accuracy (%)					Normalized Mutual Information (NMI)				
	LSI	LE	LPI	LPI-b	SSGP	LSI	LE	LPI	LPI-b	SSGP
2	0.992	0.998	0.998	0.998	**0.998**	0.965	0.981	0.981	0.981	**0.981**
3	0.985	0.996	0.996	0.996	**0.996**	0.962	0.976	0.976	0.977	**0.978**
4	0.970	0.996	0.996	**0.996**	0.995	0.942	0.979	0.979	**0.979**	0.978
5	0.961	0.993	0.993	0.993	**0.995**	0.942	0.973	0.973	**0.975**	**0.970**
6	0.954	0.992	0.992	**0.993**	0.990	0.939	0.974	0.974	**0.975**	0.968
7	0.903	0.988	0.987	**0.990**	0.989	0.892	0.966	0.968	**0.969**	0.967
8	0.890	0.987	0.988	**0.989**	0.988	0.895	0.967	0.967	**0.970**	0.968
9	0.870	0.983	0.984	0.987	**0.988**	0.878	0.967	0.966	**0.970**	0.968
10	0.850	0.978	0.979	**0.982**	0.980	0.869	0.958	0.959	**0.962**	0.972
15	0.825	0.958	0.958	0.970	**0.971**	0.844	0.942	0.946	0.961	**0.961**
20	0.810	0.924	0.923	0.954	**0.954**	0.832	0.925	0.933	0.944	**0.945**
30	0.789	0.910	0.911	0.930	**0.932**	0.822	0.915	0.916	0.926	**0.928**
Average	0.899	0.975	0.975	0.981	**0.9813**	0.898	0.96	0.961	**0.965**	0.964



(a) $\mathbf{S}_{kNN}$ (b) $\mathbf{S}_{\epsilon NN}$ (c) $\mathbf{S}_{\mathbf{C}}$

Figure 1. Affinity matrices obtained for TDT2.

- Methods from spectral approach include *Laplacian embedding* (LE) proposed in [3], *latent semantic indexing* (LSI) [8] and *locality preserving indexing* (LPI) [16]. LPI and LE construct a graph on the documents where the number of nearest neighbor is set to 15 as suggested in [16]. Two parameters are required for these algorithms: the dimension of the lower subspace r and number of clusters K. For In LPI, r is chosen as the number of nonzero singular values of data matrix $\mathbf{X}$ and K is determined from the principal subspace spanned by first $K-1$ eigenvectors of graph laplacian $\mathbf{L}$. For LSI, we use the largest K eigenvectors the covariance of the data matrix $\mathbf{X}$.
- Methods based on nonnegative matrix factorization (NMF) include *graph regularized NMF* (GNMF) [5], *symmetric NMF* [17] and *locally consistent concept factorization* (LCCF) [4]. We also compare our method with constrained semi-supervised method CITCC [23], a recent method that takes into account the constraints derived from a name identity extraction process (see [23] for details).

In our experiment, K ranges from 2 to 30 for TDT2, 2 to 25 for Reuters and 2 to 20 for 20Newsgroup data sets respectively. For a given K, we extracted 50 random subset form K permutations of all possible sets and conducted 20 tests on each subset as suggested by [16] to test the generalization of the performance.

Table II presents the clustering results obtained for TDT2 dataset comparing against various spectral approaches described earlier. This is a relatively easy dataset and, except for LSI, all methods perform well, achieving almost perfect clustering results. Our results are consistent with various performances reported earlier in [16], [5]. Though there is not much room to improve upon, on average across K, ***our proposed SSGP still achieves the best performance***. Figure 1 further illustrates the affinity matrices obtained for this

dataset. Despite having up to 30 categories, the data is well separated in the affinity matrix representation, especially the proposed method has resulted in a strong evidences of the existence of multiple subspaces in the data.

Table III and IV presents the results for Reuters dataset compared against spectral and NMF based methods. This is a more challenging dataset and our rival algorithms start to degrade in performance. ***Our results consistently outperform all of them*** in both accuracy and NMI scores. On average, our accuracy improves by 17% with respect to (w.r.t) LPI-b, 23% w.r.t LPI and LE and 30% w.r.t LSI. The NMI scores are also improved by a similar factor. For NMF-based algorithms, our proposed method improves the accuracy by 29%, 23% and 21% compared with GNMF, SymNMF and LCCF respectively. A similar improvement is recorded for NMI.

Table V reports the results for 20newsgroup dataset. This is the most challenging dataset and we compare our performance against LPI (as the best in spectral-based approach), SymNMF (as the best in NMF-based approach) and CITCC (recent state-of-the-art with extra semi-supervised information). For this dataset we use the rand index (RI) and F-measures as the performance metrics as they are often used for this dataset in the literature. Again, ***our method has resulted in a superior performance against its rivals***. On average, RI and F_β are respectively improved by 11% and 16% with respect to LPI (spectral approach); by 13% and 25% w.r.t SymNMF (NMF-based approach); and overwhelmingly by 32% and 40% w.r.t the semi-supervised method CITCC. To further illustrate how well the data is separated into subspaces, Figure 2 shows the affinity matrices. Visually, the data are well separated under our representation.

To sum up, across all datasets our proposed method has consistently resulted in better clustering performance as compared against several existing methods using different evaluation metrics. In less complex datasets (e.g., TDT2) we achieve a comparable performance,but as the data becomes more complex, the strength of our method starts to emerge and this is clearly demonstrated in Reuters and 20Newsgroup data.

C. Computational Cost Analysis

Table VI presents the computational cost in our proposed SSPG and LPI and SymNMF. The mean and standard deviation computed over 50 random subsets for a given K of Reuters data is presented. As shown, Symmetric NMF (SymNMF) is quickest; SSPG is little more expensive than SymNMF, whereas LPI is the slowest among the three methods. Interestingly, Table III and Table IV show that the LPI was the best in terms of accuracy and NMI w.r.t all other benchmark methods and specifically, SymNMF is worse than LPI by a margin of 7% in AC and 17% in NMI. SymNMF may be quickest but the performance is not on par with the current benchmark methods. While SSPG is 41-60% faster than LPI, the accuracy improves by a margin of 17% and NMI by 29%.

D. Additional Experimental Results

To further illustrate and understand the behaviour of the proposed framework, we conduct two further experiments. The first experiment is to evaluate the affine and noisy variant of our proposed method presented in Section III-A2 and Section III-A3 respectively. The second experiment is designed to test the performance our method under different graph-cut algorithms.

Table VII presents the performance on 20Newsgroup datasets for the affine and noisy data models mentioned in Sections III-A2 and III-A3 respectively. The performance is further improved for affine subspaces (SSPG-A) where the improvement in RI and F-measure is almost 2% with respect to SSPG. A similar trend is also observed for the noise model (SSPGA-N) where RI and F-measure improves by a margin of 3.6% and 5.4% .

Recall from Section II graph-cut methods (e.g., SM, ratio cut, min-max cut and NJW) are formulated on the affinity matrix $\mathbf{S}$. In following experiment, we demonstrate the performance of the graph cut methods under settings in which $\mathbf{S}$ are computed by k-NN ($\mathbf{S}_{kNN}$), ϵ-NN ($\mathbf{S}_{\epsilon NN}$) and our proposed $\mathbf{S_C}$ respectively. Figure 2 shows the affinity matrices computed on 20Newsgroup data sets for $K = 10$. Nonzero points lying outside the diagonal block of $\mathbf{S}$ deteriorates the accuracy of the clustering method. As shown in Figure 2, $\mathbf{S}_{kNN}$ and $\mathbf{S}_{\epsilon NN}$ has a noisy block structures whereas $\mathbf{S_C}$ has minimum number of nonzero points lying outside the diagonal block. Table VIII shows that the graph cut methods achieve a high performance using $\mathbf{S_C}$ with respect to $\mathbf{S}_{kNN}$ and $\mathbf{S}_{\epsilon NN}$ respectively. On average, the improvement in accuracy is close to $\frac{0.796 - 0.442}{0.442} \approx 80\%$ with respect to $\mathbf{S}_{kNN}$ and $\mathbf{S}_{\epsilon NN}$.

V. Conclusion

We have proposed a novel document clustering method that represents a document as a linear sparse combination of the remaining documents in the corpus. The sparse coefficients are learned by optimizing a l_1-regularized objective function on documents, and then a spectral algorithm is applied to group the documents into clusters. We argue that the subspaces discovered through this process naturally correspond to categories in the corpus. The novel aspect of the proposed method is the ability to learn the neighborhood structure automatically, i.e. local correlations between the documents. Unlike previous methods which compute the affinity matrix from heuristic parameter tuning, such as k-NN and ϵ-NN, our method generates an affinity matrix that automatically adapts to varying data scales. We have also extended our method for sparse representations of documents to the affine and noisy formulation, which are

Table III
PERFORMANCE ON REUTERS COMPARED AGAINST SPECTRAL-BASED APPROACHES.

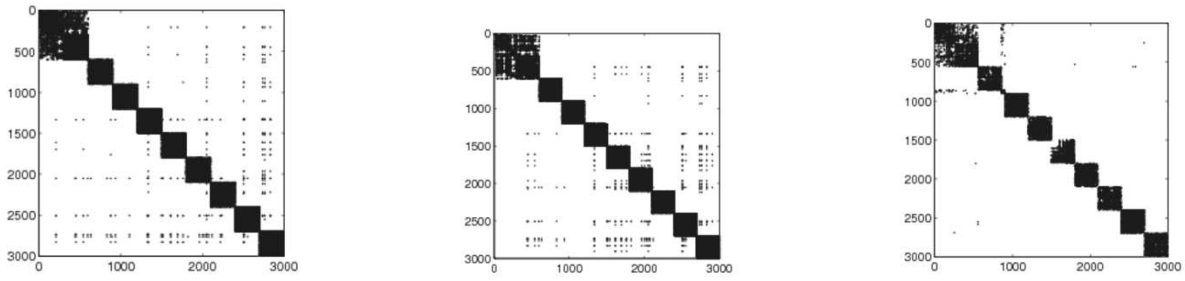
K	Accuracy (%)					Normalized Mutual Information (%)				
	LSI	LE	LPI	LPI-b	SSGP	LSI	LE	LPI	LPI-b	SSGP
2	0.864	0.923	0.923	0.963	**0.975**	0.569	0.697	0.697	0.793	**0.793**
3	0.768	0.816	0.816	0.884	**0.890**	0.536	0.601	0.601	0.660	**0.700**
4	0.715	0.793	0.793	0.843	**0.900**	0.573	0.635	0.635	0.671	**0.720**
5	0.654	0.737	0.737	0.780	**0.870**	0.538	0.603	0.603	0.633	**0.660**
6	0.642	0.719	0.719	0.760	**0.881**	0.552	0.615	0.615	0.636	**0.713**
7	0.610	0.694	0.694	0.724	**0.804**	0.547	0.617	0.617	0.629	**0.650**
8	0.572	0.650	0.650	0.693	**0.861**	0.530	0.587	0.587	0.615	**0.672**
9	0.549	0.625	0.625	0.661	**0.832**	0.532	0.586	0.586	0.605	**0.650**
10	0.540	0.615	0.615	0.646	**0.800**	0.528	0.586	0.586	0.607	**0.650**
15	0.468	0.554	0.555	0.590	**0.750**	0.492	0.548	0.549	0.560	**0.630**
20	0.461	0.475	0.474	0.511	**0.700**	0.466	0.482	0.484	0.503	**0.610**
25	0.366	0.412	0.414	0.456	**0.680**	0.356	0.417	0.416	0.454	**0.600**
Average	0.600	0.667	0.667	0.709	**0.830**	0.518	0.581	0.5813	0.613	**0.670**

Table IV
PERFORMANCE ON REUTERS COMPARED AGAINST NMF-BASED APPROACHES.

K	Accuracy (%)				Normalized Mutual Information (%)			
	GNMF	SymNMF	LCCF	SSGP	GNMF	SymNMF	LCCF	SSGP
4	0.780	0.786	0.752	**0.900**	0.620	0.533	0.556	**0.720**
6	0.690	0.701	0.677	**0.881**	0.580	0.556	0.567	**0.713**
10	0.605	0.658	0.679	**0.800**	0.541	0.510	0.506	**0.650**
20	0.456	0.509	0.601	**0.701**	0.458	0.489	0.455	**0.610**
Average	0.632	0.663	0.677	**0.820**	0.549	0.522	0.521	**0.670**

Table V
CLUSTERING RESULTS FOR 20NEWSGROUP.

K	RI				$F_{\beta=1}$			
	CITCC	SymNMF	LPI	SSGP	CITCC	SymNMF	LPI	SSGP
4	0.70	0.81	0.761	**0.89**	0.65	0.74	0.77	**0.82**
6	0.69	0.86	0.87	**0.96**	0.623	0.72	0.75	**0.88**
12	0.65	0.76	0.81	**0.91**	0.611	0.66	0.71	**0.93**
20	0.58	0.68	0.67	**0.80**	0.545	0.58	0.65	**0.86**
Average	0.655	0.77	0.78	**0.89**	0.60	0.67	0.72	**0.87**



(a) $\mathbf{S}_{kNN}$ (b) $\mathbf{S}_{\epsilon NN}$ (c) $\mathbf{S}_{\mathbf{C}}$

Figure 2. Affinity matrices obtained for 20Newsgroup for $K = 10$.

Table VI
COMPUTATIONAL TIME IN SECONDS: ON REUTERS DATA

K	SymNMF		LPI		SSPG	
	Mean	Median	Mean	Median	Mean	Median
4	**4.61**	**0.13**	67	4.39	39	9
8	**39**	**5.85**	1079	1011	82	39
10	**5**	**5.70**	1248	963	144	34

Table VII
PERFORMANCE ON 20NEWSGROUP USING A VARIANT OF OUR METHOD IN WHICH AFFINE SUBSPACES AND HANDLING DATA NOISES PRESENTED IN SECTION III-A2 AND SECTION III-A3 IS USED.

K	RI			F_β		
	SSPG	SSPG-A	SSPG-N	SSPG	SSPG-A	SSPG-N
4	0.885	0.897	**0.900**	0.820	0.849	**0.870**
6	0.956	0.968	**0.971**	0.880	0.900	**0.941**
12	0.910	0.927	**0.941**	0.930	0.941	**0.965**
20	0.800	0.842	**0.860**	0.860	0.881	**0.901**
Average	0.887	0.900	**0.918**	0.872	0.885	**0.919**

Table VIII
GRAPH CUT METHODS: PERFORMANCE ON 20 NEWSGROUP DATA

Methods	Accuracy (%)		
	$\mathbf{S}_{kNN}$	$\mathbf{S}_{\epsilon NN}$	$\mathbf{S_C}$
SM method [22]	0.340	0.344	**0.767**
Ratio Cut [11]	0.330	0.331	**0.756**
Min-max Cut [12]	0.450	0.480	**0.761**
NJW method [20]	0.650	0.640	**0.900**
Average	0.442	0.448	**0.796**

demonstrated to even provide better results. We validated our results by conducting intensive experiments on three real-world news datasets and showed that its performance is clearly superior to current state-of-the-arts, including LSI[8], LPI[16], NMF[17] and semi-supervised algorithm CITCC[23].

APPENDIX

ALM UPDATES FOR NOISY FORMULATION

In what follows, we derive the iterative updates for $\mathbf{a}$ and y in (15). The principle of the alternating method is to fix one set of variables and solve for the others, and repeat until convergence is found. When we fix $\mathbf{a}$ and solve for y, the ALM update is standard as follows

$$y^{k+1} = y^k + \eta(1 - \mathbf{1^T a^k}).$$

Here, the superscript denotes the iteration number. Next, we fix y and solve for $\mathbf{a}$. The objective function with respect to $\mathbf{a}$ is

$$\begin{aligned}\mathcal{L}(\mathbf{a}) &= \|\mathbf{a}\|_1 + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{X}^{-i}\mathbf{a}\|_2^2 \\ &\quad + y(1 - \mathbf{1}^T\mathbf{a}) + \frac{\eta}{2}(1 - \mathbf{1}^T\mathbf{a})^2 \\ &= \|\mathbf{a}\|_1 + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{X}^{-i}\mathbf{a}\|_2^2 \\ &\quad + \frac{\eta}{2}\mathbf{a}^T\mathbf{1}\mathbf{1}^T\mathbf{a} - \eta\mathbf{a}^T\mathbf{1} + \text{const.} \qquad (16)\end{aligned}$$

We show that it is possible to convert this objective function to a Lasso form. In fact, expanding the objective function we obtain

$$\begin{aligned}\mathcal{L}(\mathbf{a}) &= \|\mathbf{a}\|_1 + \text{const} \\ &\quad + \frac{\mu}{2}\left(\mathbf{a}^T{\mathbf{X}^{-i}}^T\mathbf{X}^{-i}\mathbf{a} + \frac{\eta}{\mu}\mathbf{a}^T\mathbf{1}\mathbf{1}^T\mathbf{a}\right) \\ &\quad - \frac{\mu}{2}\left(2\mathbf{a}^T\left({\mathbf{X}^{-i}}^T\mathbf{x}_i + \frac{(\eta + y)\mathbf{1}}{\mu}\right)\right). \qquad (17)\end{aligned}$$

To simplify the notation, we define

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Y} \\ (\eta/\mu)\mathbf{1}^T \end{bmatrix}, \quad (18)$$

$$\mathbf{c} = \mathbf{X}^{-i^T}\mathbf{x}_i + \frac{(\eta+y)\mathbf{1}}{\mu}. \quad (19)$$

Then the objective function is written as

$$\mathcal{L}(\mathbf{a}) = \|\mathbf{a}\|_1 + \frac{\mu}{2}\left(\mathbf{a}^T\mathbf{Z}^T\mathbf{Z}\mathbf{a} - 2\mathbf{a}^T\mathbf{c}\right) + \text{const.} \quad (20)$$

Let $\mathbf{m}$ be a vector such that $\mathbf{Z}^T\mathbf{m} = \mathbf{c}$, then we can write

$$\mathcal{L}(\mathbf{a}) = \|\mathbf{a}\|_1 + \frac{\mu}{2}\|\mathbf{m} - \mathbf{Z}\mathbf{a}\|_2^2 + \text{const.} \quad (21)$$

This is the Lasso form and thus can be solved with many efficient Lasso-type optimization packages.

REFERENCES

[1] "Text datasets in matlab format." [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

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

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Proc. NIPS*, vol. 14, pp. 585–591, 2001.

[4] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, no. 99, pp. 1–1, 2011.

[5] D. Cai, X. He, J. Han, and T. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[6] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, 2011.

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

[8] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[9] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. ACM SIGKDD*, 2001, pp. 269–274.

[10] I. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proc. ACM SIGKDD*. ACM, 2004, pp. 551–556.

[11] ——, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[12] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. ICDM*, 2001, pp. 107–114.

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

[14] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. CVPR*. Ieee, 2009, pp. 2790–2797.

[15] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *Proc. ACM SIGKDD*, 2009, pp. 359–368.

[16] X. He, D. Cai, H. Liu, and W. Ma, "Locality preserving indexing for document representation," in *Proc. ACM SIGIR*, 2004, pp. 96–103.

[17] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SDM*, 2012.

[18] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[19] M. Meila, "Comparing clusterings: an axiomatic view," in *Proc. ICML*, 2005, pp. 577–584.

[20] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Proc. NIPS*, vol. 2, pp. 849–856, 2002.

[21] Y. Ni, J. Sun, X. Yuan, S. Yan, and L. Cheong, "Robust low-rank subspace segmentation with semidefinite guarantees," in *Proc. ICDMW*, 2010, pp. 1179–1188.

[22] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[23] Y. Song, S. Pan, S. Liu, F. Wei, M. Zhou, and W. Qian, "Constrained co-clustering for textual documents," in *Proc. AAAI*, 2010.

[24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[25] S. Wang, X. Yuan, T. Yao, S. Yan, and J. Shen, "Efficient subspace segmentation via quadratic programming," in *Proc. AAAI*, 2011.

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

[27] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. CVPR*. Ieee, 2008, pp. 1–8.

[28] Y. Yu and D. Schuurmans, "Rank/norm regularization with closed-form solutions: Application to subspace clustering," *Arxiv preprint arXiv:1202.3772*, 2012.

[29] H. Zha, C. Ding, M. Gu, X. He, and H. Simon, "Spectral relaxation for k-means clustering," *Proc. NIPS*, vol. 14, pp. 1057–1064, 2001.

[30] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proc. ICIKM*. ACM, 2001, pp. 25–32.