Spectral Clustering with Automatic Cluster-Number Identification via Finding Sparse Eigenvectors

Yuto Ogino*, Masahiro Yukawa*†
*Dept. Electronics and Electrical Engineering, Keio University, Japan
†Center for Advanced Intelligence Project, RIKEN, Japan

Abstract—**Spectral clustering is an empirically successful approach to separating a dataset into some groups with possibly complex shapes based on pairwise affinity. Identifying the number of clusters automatically is still an open issue, although many heuristics have been proposed. In this paper, imposing sparsity on the eigenvectors of graph Laplacian is proposed to attain reasonable approximations of the so-called cluster-indicator-vectors, from which the clusters as well as the cluster number are identified. The proposed algorithm enjoys low computational complexity as it only computes a relevant subset of eigenvectors. It also enjoys better clustering quality than the existing methods, as shown by simulations using nine real datasets.**

I. Introduction

Clustering is an unsupervised machine learning task of great importance, aiming to group a given dataset based on some affinity measure. The important problem involved with clustering is how to determine the number of clusters. Although several methods for the automatic determination have been proposed (e.g., [1], [2], [3]), most (if not all) of them suffer from high computational costs and/or low accuracy. Spectral clustering [4], [5], [6] is one of the clustering methods, which exploits the so-called graph-Laplacian matrix. It is known to be able to identify clusters with non-convex boundaries, while k-means [7], [8] tends to yield spherical clusters. The eigenvectors of a graph Laplacian are some linear combinations of the indicator vectors, where the indices of the nonzero components indicate the clusters. If the indicator vectors themselves are attained instead of their linear combinations, the clusters and its number can readily be obtained. Self-tuning spectral clustering (STSC) [1] tries to decouple the linear combinations via certain optimization problem. As mentioned therein, however, the cost function used is motivated by simplicity and perfectible. It has indeed been reported that STSC often gives low clustering accuracy due to underestimation of the cluster number [3]

In this paper, we propose an efficient spectral clustering algorithm which identifies the cluster number automatically. The proposed algorithm seeks to obtain the indicator vectors in a way different from STSC. Since the indicator vectors of the clusters are sparsest among their linear combinations, the eigenproblem is formulated with a sparse regularizer. The proposed algorithm computes the sparse eigenvectors and the corresponding clusters successively using an eigenvalue deflation method [9], until each data point is assigned to one of the clusters. The number of sparse eigenvectors obtained is thus no more than the cluster number, and this saves the computational time. The simulation results show that the proposed algorithm tends to attain better clustering quality than the existing ones for real datasets in terms of normalized mutual information (NMI) [10], [11].

Notation

Let $\mathbb{S}_+^n$ denote the set of n-dimensional symmetric positive semi-definite matrices. Define $\overline{1,n} := \{1, 2, \cdots, n\}$. Given $\mathcal{A} \subseteq \overline{1,n}$, define the *indicator vector* $\mathbf{1}_\mathcal{A} \in \mathbb{R}^n$ by

$$\forall i \in \overline{1,n}: \quad (\mathbf{1}_\mathcal{A})_i := \begin{cases} 1 & i \in \mathcal{A} \\ 0 & i \notin \mathcal{A}. \end{cases} \tag{1}$$

Define $\mathbf{1} := \mathbf{1}_{\overline{1,n}}$. Given $\boldsymbol{a} \in \mathbb{R}^n$, let $\operatorname{diag}(\boldsymbol{a})$ denote the diagonal matrix with entries given by $\boldsymbol{a}$. Define the *support* $\operatorname{supp}(\boldsymbol{a}) := \left\{ i \in \overline{1,n} | a_i \neq 0 \right\}$. Define the ℓ_0 *pseudo-norm* $\|\boldsymbol{a}\|_0 := |\operatorname{supp}(\boldsymbol{a})|$. The term *i-th principal (minor) eigenvector* refers to the eigenvector associated with the eigenvalue λ_i where $|\lambda_j| \geq |\lambda_k|$ ($|\lambda_j| \leq |\lambda_k|$) for all $j < k$.

II. Preliminaries

We present the minimum knowledge required to understand the proposed algorithm. See e.g. [6] for the comprehensive tutorial on spectral clustering. Spectral clustering takes a dataset $\{x_1, \cdots, x_n\}$ and the corresponding affinity matrix $\boldsymbol{W} \in \mathbb{R}_+^{n \times n}$ whose entry $w_{i,j}\, (= w_{j,i})$ represents the affinity between x_i and x_j. Consider a partition $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$ of the indices $\overline{1,n}$ as clusters of the dataset. Let $\sim$ denote the equivalence relation with respect to $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$, i.e.

$$i \sim j \iff \left(\exists k \in \overline{1,c} \quad \text{s.t.} \quad i, j \in \mathcal{C}_k \right). \tag{2}$$

Spectral clustering then seeks the clusters $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$ such that:

- *intra-cluster affinity* ($w_{i,j}$ for $i \sim j$) is large.
- *inter-clusters affinity* ($w_{i,j}$ for $i \nsim j$) is small.

Spectral clustering exploits a matrix called *graph Laplacian*, for which several definitions exist:

$$\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W} \tag{3}$$

$$\boldsymbol{L}_{\text{sym}} := \boldsymbol{I} - \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2} \tag{4}$$

$$\boldsymbol{L}_{\text{rw}} := \boldsymbol{I} - \boldsymbol{D}^{-1} \boldsymbol{W}, \tag{5}$$

This work was supported in part by JSPS Grants-in-Aids (15K06081, 15K13986, 15H02757).

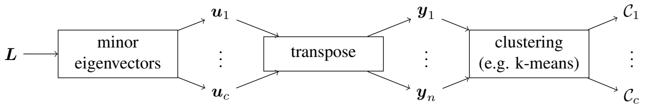


Fig. 1. The conceptual diagram of the classical spectral clustering.

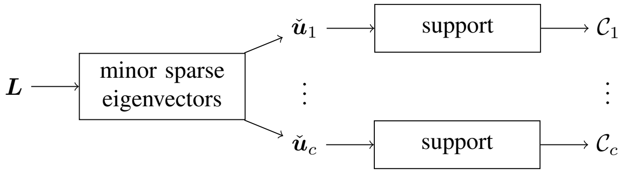

$\boldsymbol{L}$
minor sparse eigenvectors
$\check{\boldsymbol{u}}_1$
$\vdots$
$\check{\boldsymbol{u}}_c$
support
support
$\mathcal{C}_1$
$\vdots$
$\mathcal{C}_c$

Fig. 2. The conceptual diagram of the proposed algorithm.

where $\boldsymbol{D} := \operatorname{diag}(\boldsymbol{W}\boldsymbol{1})$. Since $\boldsymbol{W}$ is symmetric and non-negative, the followings hold:

Proposition 1 (Properties of the graph Laplacians [6]). *(a) All eigenvalues of $\boldsymbol{L}$, $\boldsymbol{L}_{\mathrm{sym}}$ and $\boldsymbol{L}_{\mathrm{rw}}$ are non-negative.*

(b) $\boldsymbol{L}_{\mathrm{sym}}\boldsymbol{v} = \lambda\boldsymbol{v} \iff \boldsymbol{L}_{\mathrm{rw}}\left(\boldsymbol{D}^{-1/2}\boldsymbol{v}\right) = \lambda\left(\boldsymbol{D}^{-1/2}\boldsymbol{v}\right)$

(c) Let $\mathcal{G}$ be a graph whose weighted adjacency matrix is $\boldsymbol{W}$, and $\tilde{\mathcal{C}}_i \subseteq \overline{1,n}$ $\left(\forall i \in \overline{1,m}\right)$ be the connected component of $\mathcal{G}$. Then $\boldsymbol{L}$ and $\boldsymbol{L}_{\mathrm{rw}}$ share the same eigenspace $\operatorname{span}\left\{\boldsymbol{1}_{\tilde{\mathcal{C}}_1}, \cdots, \boldsymbol{1}_{\tilde{\mathcal{C}}_m}\right\}$ corresponding to the eigenvalue 0.

We now consider the ideal case where the desired clusters $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$ have exactly zero inter-cluster affinity:

$$\forall i, j \in \overline{1,n}\colon \quad i \nsim j \Longrightarrow w_{i,j} = 0. \tag{6}$$

The clusters $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$ then coincide with the connected components $\left\{\tilde{\mathcal{C}}_1, \cdots, \tilde{\mathcal{C}}_m\right\}$ in Proposition 1. Therefore the minor eigenvectors $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_c\}$ of $\boldsymbol{L}$ give a basis of $\operatorname{span}\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$. Define $\boldsymbol{U} := \left[\begin{array}{ccc} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_c \end{array}\right] \in \mathbb{R}^{n\times c}$ and let $\boldsymbol{y}_i \in \mathbb{R}^c$ be the i-th column of $\boldsymbol{U}^\top$. The following property is useful to identify $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$:

$$\forall i, j \in \overline{1,n}\colon \quad i \sim j \iff \boldsymbol{y}_i = \boldsymbol{y}_j. \tag{7}$$

In more practical settings, the inter-cluster affinity is not necessarily zero, but takes small values. If one decomposes $\boldsymbol{L}$ into an intra-cluster (inter-cluster) term $\hat{\boldsymbol{L}}$ ($\boldsymbol{E}$):

$$\boldsymbol{L} = \hat{\boldsymbol{L}} + \boldsymbol{E}, \tag{8}$$

$$\hat{l}_{i,j} := \begin{cases} l_{i,j} & i \sim j \\ 0 & i \nsim j \end{cases}, \quad e_{i,j} := \begin{cases} 0 & i \sim j \\ l_{i,j}\,(= -w_{i,j}) & i \nsim j \end{cases}, \tag{9}$$

then the minor eigenvectors of $\hat{\boldsymbol{L}}$ form a basis of $\operatorname{span}\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$ as in the ideal case mentioned above. Furthermore, since $\boldsymbol{E}$ has a small Frobenius norm, the eigenvectors of $\boldsymbol{L}$ are close to those of $\hat{\boldsymbol{L}}$ in a certain sense, according to the matrix perturbation theory [12]. This implies that (7) approximately holds, and the clusters $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$ can be identified by clustering $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ as illustrated in Figure 1.

III. Proposed algorithm

We present an efficient spectral clustering algorithm without assuming given c. The key problem of the spectral clustering is that $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_c\}$ can be an arbitrary orthonormal basis of $\operatorname{span}\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$, and there is no guarantee that it coincides with $\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$. This complicates the subsequent clustering process, and thus one cannot compute $\mathcal{C}_i$ sequentially as i increases. We employ the sparse regularization to solve this issue.

A. Main idea

The proposed algorithm uses the sparse regularization to compute the minor eigenvectors $\{\check{\boldsymbol{u}}_1, \cdots, \check{\boldsymbol{u}}_c\}$ of the graph Laplacian $\boldsymbol{L}$. The regularization leads to obtaining the sparsest vectors, i.e. $\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$, in the eigenspace $\operatorname{span}\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$. Then each cluster $\mathcal{C}_i$ can be identified independently by taking the support of $\check{\boldsymbol{u}}_i$, as depicted in Figure 2. Only a relevant subset of eigenvectors needs to be computed since the algorithm terminates once all the data points are assigned to the clusters.

It is well known in basic linear algebra that for all $\boldsymbol{A} \in \mathbb{S}^n$, a first principal eigenvector $\boldsymbol{u}(\boldsymbol{A})$ is given by

$$\boldsymbol{u}(\boldsymbol{A}) \in \underset{\boldsymbol{u}\in\mathcal{S}^n}{\operatorname{argmax}}\ \boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}, \tag{10}$$

where $\mathcal{S}^n := \left\{\boldsymbol{u} \in \mathbb{R}^n \colon \boldsymbol{u}^\top\boldsymbol{u} = 1\right\}$ is the unit sphere. We define a *first principal sparse eigenvector*:

$$\check{\boldsymbol{u}}(\boldsymbol{A}) \in \underset{\boldsymbol{u}\in\mathcal{S}^n}{\operatorname{argmax}}\left(\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u} - \rho\,\|\boldsymbol{u}\|_0\right), \tag{11}$$

where $\rho > 0$ is the sparsity-controlling parameter. Several feasible algorithms have been proposed to approximate $\check{\boldsymbol{u}}(\boldsymbol{A})$ for $\boldsymbol{A} \in \mathbb{S}^n_+$, mainly in the context of sparse principal component analysis (SPCA) [13], [14], [15], [16].

A *first minor sparse eigenvector* $\check{\boldsymbol{u}}_1$ of $\boldsymbol{L}$, which is needed for the proposed algorithm, is defined as $\check{\boldsymbol{u}}_1 := \check{\boldsymbol{u}}(-\boldsymbol{L}) = \check{\boldsymbol{u}}(\boldsymbol{L}_{\mathrm{rev}})$, where $\boldsymbol{L}_{\mathrm{rev}} := \lambda_{\max}\boldsymbol{I} - \boldsymbol{L} \in \mathbb{S}^n_+$ and $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{L}$. The subsequent minor sparse eigenvectors $\{\check{\boldsymbol{u}}_2, \cdots, \check{\boldsymbol{u}}_c\}$ can be obtained with one of the eigenvalue deflation methods.

B. Implementation details

1) Choice of the graph Laplacian: The use of the unnormalized Laplacian $\boldsymbol{L}$ tends to give poor results. Using normalized Laplacian $\boldsymbol{L}_{\mathrm{rw}}$ or $\boldsymbol{L}_{\mathrm{sym}}$ is therefore suggested in the proposed algorithm. Furthermore, $\boldsymbol{L}_{\mathrm{rw}}$ is better since Proposition 1 (c), which does not apply to $\boldsymbol{L}_{\mathrm{sym}}$, is the key to our derivation. The sparse eigenvectors of $\boldsymbol{L}_{\mathrm{rw}}$ however cannot be computed directly due to its asymmetry. Hence the proposed algorithm computes a minor sparse eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{L}_{\mathrm{sym}}$, and then regard $\boldsymbol{D}^{-1/2}\boldsymbol{v}_i$ as a sparse eigenvector of $\boldsymbol{L}_{\mathrm{rw}}$.

2) Effects of the noise: In the practical settings where the inter-cluster affinity is not exactly zero, $\{\check{\boldsymbol{u}}_1, \cdots, \check{\boldsymbol{u}}_c\}$ are not exactly $\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$. This causes the following two issues:

(a) $\operatorname{supp}(\check{\boldsymbol{u}}_i)$ is a bad estimate of $\mathcal{C}_i$ since $\check{\boldsymbol{u}}_i$ is likely to have very small entries such as 10^{-5}.

(b) The resulting clusters may overlap.

ISBN 978-90-827970-1-5 © EURASIP 2018

To deal with issue (a), we use the following function instead of supp:

$$\operatorname{supp}'_r\colon \boldsymbol{x} \mapsto \left\{ i \in \overline{1,n}\colon\ |x_i| > r \max_{j\in\overline{1,n}} |x_j| \right\}, \tag{12}$$

where $0 < r < 1$ is a parameter. As a remedy for issue (b), $\mathcal{C}_i$ is estimated by

$$\mathcal{C}_i \coloneqq \mathcal{R}_{i-1} \cap \operatorname{supp}'_r(\check{\boldsymbol{u}}_i), \tag{13}$$

where $\mathcal{R}_i \coloneqq \mathcal{R}_{i-1} \backslash \mathcal{C}_i \left(\mathcal{R}_0 \coloneqq \overline{1,n}\right)$ is the set of indices which are not yet assigned to any clusters at the i-th iteration.

3) Parameters: The sparsity-controlling parameter $\rho > 0$ needs to be tuned appropriately since the solution of (11) approaches that of (10) as $\rho \to 0$, while it becomes too sparse as ρ is too large. The optimization problem (11) is often relaxed using the ℓ_1-norm for tractability. Observe the following equation:

$$\max_{\boldsymbol{u}\in\mathcal{S}^n} \|\boldsymbol{u}\|_1 = \sqrt{n}. \tag{14}$$

This implies that the "magnitude" of the ℓ_1-term depends on the number n of data points, and that $\rho' \coloneqq \sqrt{n}\rho$ should be less dependent on the data. In fact, the fixed $\rho' = 0.1$ gives reasonable performances in our experiments (excluding one dataset; see Table I).

Another parameter is $0 < r < 1$ for the threshold of supp'_r. The fixed $r = 0.1$ empirically gives good performances.

The proposed method is summarized in Algorithm 1.

Algorithm 1 Sparse Spectral Clustering (proposed)

```
Input: The affinity matrix $\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$
Parameter: $\rho' > 0$
$r \leftarrow 0.1$ and $\rho \leftarrow \rho'/\sqrt{n}$
$\boldsymbol{D} \leftarrow \operatorname{diag}(\boldsymbol{W}\boldsymbol{1})$
$\boldsymbol{L}_{\mathrm{sym}} \leftarrow \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$
$\lambda_{\max} \leftarrow$ (the largest eigenvalue of $\boldsymbol{L}_{\mathrm{sym}}$)
$\boldsymbol{A}_0 \leftarrow \lambda_{\max}\boldsymbol{I} - \boldsymbol{L}_{\mathrm{sym}}$
$i \leftarrow 0$ and $\mathcal{R}_0 \leftarrow \overline{1,n}$
repeat
    $\boldsymbol{v}_i \leftarrow \operatorname{argmax}_{\boldsymbol{v}\in\mathcal{S}^n} \left(\boldsymbol{v}^\top \boldsymbol{A}_i \boldsymbol{v} - \rho\|\boldsymbol{v}\|_0\right)$ (approximately)
    $\boldsymbol{A}_{i+1} \leftarrow \boldsymbol{A}_i - \left(\boldsymbol{v}_i^\top \boldsymbol{A}_i \boldsymbol{v}_i\right)\boldsymbol{v}_i\boldsymbol{v}_i^\top$
    $\check{\boldsymbol{u}}_i \leftarrow \boldsymbol{D}^{-1/2}\boldsymbol{v}_i$
    $\mathcal{C}_i \leftarrow \mathcal{R}_{i-1} \cap \operatorname{supp}'_r(\check{\boldsymbol{u}}_i)$
    $\mathcal{R}_i \leftarrow \mathcal{R}_{i-1} \backslash \mathcal{C}_i$
    $i \leftarrow i + 1$
until $\mathcal{C}_i = \emptyset$
Output: $\mathcal{C}_1, \cdots, \mathcal{C}_{i-1}$ and $\mathcal{R}_i$
```

IV. Simulation Results

Numerical experiments are conducted to validate the following claims:

(a) The minor sparse eigenvectors of the graph Laplacian approximate the cluster-indicating vectors.

(b) The proposed algorithm gives visually natural clustering results.

(c) The proposed algorithm outperforms the existing methods in terms of clustering quality.

A. Synthetic datasets

1) Settings: Two datasets $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^2$ are used. The first data model is a mixture of four well-separated Gaussians, each of which has mean $(\pm 5, \pm 5) \in \mathbb{R}^2$ and covariance $\boldsymbol{I}$. 150 points are i.i.d. sampled from each Gaussian, resulting $n = 600$ points in total. The affinity matrix $\boldsymbol{W}$ is then set to the Gaussian kernel matrix $\boldsymbol{K}$:

$$k_{i,j} \coloneqq \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) \tag{15}$$

with $\sigma = 1$.

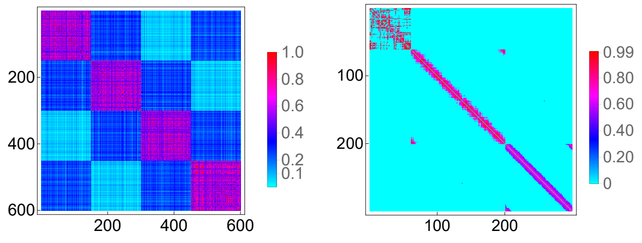
The second dataset[1] consists of three concentric circles with 62, 99, 138 data points, respectively ($n = 299$). For this data, $\boldsymbol{W}$ is designed as follows:

$$\boldsymbol{W} \coloneqq \boldsymbol{K} \odot \boldsymbol{A}, \tag{16}$$

Where $\boldsymbol{A}$ is the adjacency matrix of the k-nearest neighbor graph of the dataset, and $\odot$ denotes the Hadamard product. The parameters are set to $\sigma = 1$, $k = 5$.

To compute the sparse eigenvectors, iterative minimization of rectangular procrustes (IMRP) [16] is used along with Hotelling's deflation. The parameters for the surrogate function of IMRP is fixed to $p = 1$, $\epsilon = 10^{-3}$.

2) Results: Figure 3 shows the affinity matrix $\boldsymbol{W}$ for each dataset, where the indices are sorted according to the clusters. One can see that $\boldsymbol{W}$ of both cases are approximately block-diagonal. This indicates that the inter-cluster affinity is small. Figure 4 shows the sparse eigenvectors $\{\check{\boldsymbol{u}}_1, \cdots, \check{\boldsymbol{u}}_c\}$ generated by Algorithm 1 for the Gaussian mixture dataset. The vectors $\{\check{\boldsymbol{u}}_1, \cdots, \check{\boldsymbol{u}}_c\}$ approximate the cluster-indicating vectors $\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$, verifying the claim (a). Furthermore, $\{\check{\boldsymbol{u}}_1, \cdots, \check{\boldsymbol{u}}_c\}$ for the circles dataset were almost exactly equal to $\{\boldsymbol{1}_{\mathcal{C}_1}, \cdots, \boldsymbol{1}_{\mathcal{C}_c}\}$ (the figure is omitted due to the lack of space). Consequently the proposed algorithm identifies the correct number of clusters for these datasets. To support the claim (b), the identified clusters are depicted in Figure 5 with different colors.

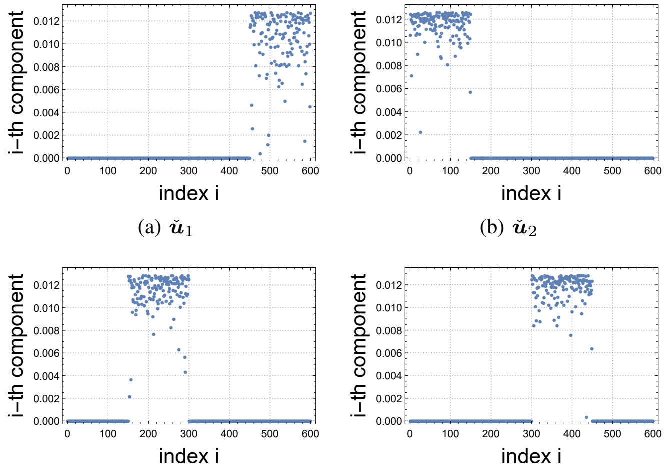


(a) mixture of Gaussians

(b) concentric circles

Fig. 3. The affinity matrix $\boldsymbol{W}$.

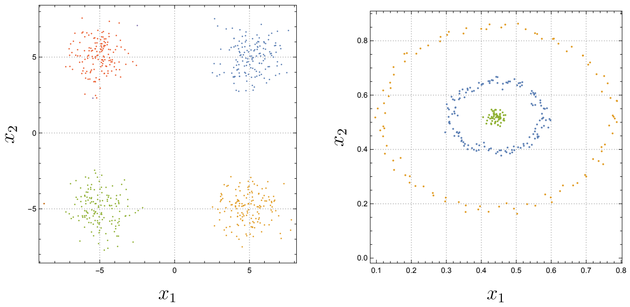
[1] Available at www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html



(a) $\tilde{\boldsymbol{u}}_1$ (b) $\tilde{\boldsymbol{u}}_2$

(c) $\tilde{\boldsymbol{u}}_3$ (d) $\tilde{\boldsymbol{u}}_4$

Fig. 4. The sparse eigenvectors $\tilde{\boldsymbol{u}}_1, \cdots, \tilde{\boldsymbol{u}}_4$ for the mixture of Gaussians dataset.



(a) mixture of Gaussians (b) concentric circles

Fig. 5. The clustering results of the 2D data points.

B. Real datasets

1) Settings: The clustering quality of the proposed algorithm was compared to the existing ones, using real datasets for the classification. The datasets are exactly the same as in the numerical experiments conducted in [3]. The affinity matrices are again constructed via the k-nearest neighbor graph and the Gaussian kernel (with parameter σ). Some of the datasets are standardized beforehand so as to have zero mean and unit variances. Table I shows the sample size n, dimension d of data, the parameters σ, k for constructing $\boldsymbol{W}$, and the sparsity-inducing parameter ρ' used in Algorithm 1.

TABLE I
ATTRIBUTES AND PARAMETERS FOR EACH DATASET

dataset	n	d	σ	k	ρ'	standardized
Opt. Digits.	5620	64	100	5	0.10	No
Pen Digits	10992	16	50	5	0.10	No
M.F. Digits	2000	216	10	5	0.10	Yes
Satellite	6435	36	30	5	0.10	No
Yale Faces	5850	1200	1000	5	0.10	No
Phoneme	4509	256	50	5	0.10	No
Smartphone	10929	561	30	5	0.06	Yes
ISOLET	6238	617	30	5	0.10	Yes
Image Seg.	2310	19	50	50	0.10	Yes

The similarity between the true labels and the clustering results are measured based on *normalized mutual information* (NMI). NMI between two random variables X, Y is defined as follows:

$$\mathrm{NMI}\left(X,Y\right) := \frac{I\left(X,Y\right)}{\sqrt{H\left(X\right)H\left(Y\right)}}, \tag{17}$$

where $I\left(X,Y\right) := H\left(X\right) - H\left(X|Y\right) = H\left(Y\right) - H\left(Y|X\right)$ is the mutual information and H is the Shannon entropy:

$$H\left(X\right) := \mathbb{E}_X\left[-\log_2 p\left(X\right)\right] \tag{18}$$

$$H\left(X|Y\right) := \mathbb{E}_{X,Y}\left[-\log_2 p\left(X|Y\right)\right]. \tag{19}$$

In our case X and Y correspond to the true label and the estimated label for each data point. Given true clusters $\{\mathcal{C}_1^\star, \cdots, \mathcal{C}_{c^\star}^\star\}$ and estimates $\{\mathcal{C}_1, \cdots, \mathcal{C}_c\}$, the empirical joint probabilities are defined as

$$\forall x \in \overline{1, c^\star},\ y \in \overline{1, c}: \quad p\left(x,y\right) := \left|\mathcal{C}_x^\star \cap \mathcal{C}_y\right|/n. \tag{20}$$

The following hold for all random variables X, Y:

- $0 \leq \mathrm{NMI}\left(X,Y\right) \leq 1$;
- $\mathrm{NMI}\left(X,Y\right) = 0 \iff X, Y$ are independent;
- $\mathrm{NMI}\left(X,Y\right) = 1 \iff$ knowing X determines Y, and vice versa.

A higher NMI thus implies that X, Y are more mutually-dependent (i.e. the estimated clusters are more accurate).

2) Results: The true number $c^\star$ of clusters, the estimated number c, and NMI are summarized in Table II. Experimental results for the following clustering algorithms are taken from [3]: spectral partitioning using density separation (SPUD) [3], self-tuning spectral clustering (STSC) [1], spectral clustering using cluster distortion (SCCD)[2] [5], alternative spectral clustering (Alt. SC)[3], k-means [8] with gap statistic [17] to estimate c, optimal extraction of clusters from hierarchies (OCE) [18], Gaussian mixture model (GMM) [19] with Bayesian information criterion to estimate c, DBSCAN [20]. A brief summary of all these algorithms can be found in [3]. Note that the affinity matrix $\boldsymbol{W}$ for the proposed algorithm is sparsified for computational efficiency by using the kernel matrix $\boldsymbol{K}$ and the k-nearest neighbor graph, whereas it was set to the kernel matrix $\boldsymbol{K}$ itself in [3].

Table II shows that the proposed algorithm outperforms all the other algorithms on average. In addition, it often achieves the highest or nearly highest NMI for each dataset.

The datasets with NMI less than 0.7 are further investigated. Figures 6–9 illustrate the matrix $\boldsymbol{W}$ for these datasets. The indices of $\boldsymbol{W}$ in each subfigure is sorted according to the true clusters or the clusters estimated by the proposed algorithm, respectively. Figures 6(b)–9(b) are nearly block-diagonal, indicating that the inter-cluster affinity for the estimated clusters are small. This suggests that the proposed algorithm gives reasonable results for the given $\boldsymbol{W}$.

[2] SCCD cannot identify the cluster number.

[3] P. Bruneau, "speccalt: Alternative spectral clustering, with automatic estimation of k." 2013.

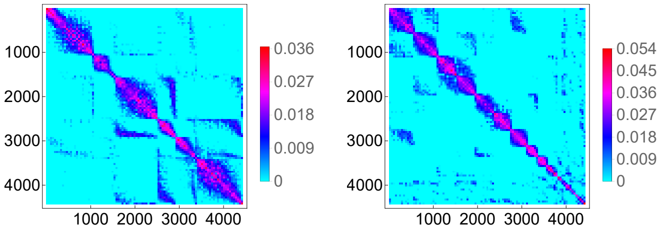
ISBN 978-90-827970-1-5 © EURASIP 2018

V. Conclusion

We proposed an efficient spectral clustering algorithm that identifies the cluster number automatically. The proposed algorithm was derived by formulating the clustering task as a sparse eigenvalue problem for the graph Laplacian matrix. It computes the sparse minor eigenvector recursively until each data point is assigned to one of the clusters, thus being computationally efficient. The simulation results demonstrated that the proposed algorithm attained significantly higher average-NMI than the existing methods over the real benchmark datasets.

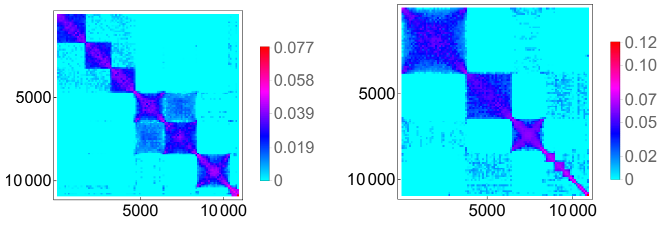
TABLE II
Clustering quality based on NMI for benchmark datasets. The results of other algorithms were taken from [3]. The highest NMI for each dataset is highlighted in bold.

dataset \ algorithm		proposed	SPUDS	STSC	SCCD	Alt.SC	k-means	OCE	GMM	DBSCAN
Yale Faces	NMI	**0.92**	0.84	0.04	0.73	0.28	0.70	0.81	0.78	0.54
$(c^\star = 10)$	c	16	14	2	-	7	20	10	9	47
Opt. Digits	NMI	**0.86**	0.79	0.73	0.01	0.01	0.68	0.59	0.63	0.56
$(c^\star = 10)$	c	11	13	9	-	3	19	4	9	8
Phoneme	NMI	0.86	0.69	0.66	**0.87**	0.63	0.68	0.68	0.61	0.37
$(c^\star = 5)$	c	5	7	3	-	3	10	3	4	3
M.F. Digits	NMI	**0.83**	0.79	0.71	0.80	0.73	0.72	0.57	0.00	0.65
$(c^\star = 10)$	c	16	18	2	-	19	20	5	9	27
Pen Digits	NMI	0.77	0.70	0.38	0.19	**0.80**	0.73	0.64	0.73	0.69
$(c^\star = 10)$	c	12	9	2	-	19	20	5	9	27
ISOLET	NMI	0.66	**0.72**	0.64	0.70	0.26	0.70	0.42	0.40	0.25
$(c^\star = 26)$	c	10	25	15	-	2	52	2	2	5
Smartphone	NMI	**0.65**	0.55	0.57	0.51	0.49	0.56	0.57	0.00	0.41
$(c^\star = 12)$	c	19	7	2	-	2	24	2	1	3
Image Seg.	NMI	0.61	**0.68**	0.42	0.03	0.01	0.61	0.46	0.62	0.50
$(c^\star = 7)$	c	9	6	3	-	3	14	2	8	5
Satellite	NMI	0.60	0.40	0.39	0.62	**0.66**	0.60	0.38	0.55	0.51
$(c^\star = 10)$	c	27	3	2	-	7	11	3	9	5
Average NMI		**0.75**	0.68	0.50	0.50	0.43	0.66	0.57	0.48	0.50



(a) sorted by true clusters (b) sorted by estimated clusters

Fig. 6. The affinity matrices $\boldsymbol{W}$ for Satellite dataset $(c^\star = 10)$.



(a) sorted by true clusters (b) sorted by estimated clusters

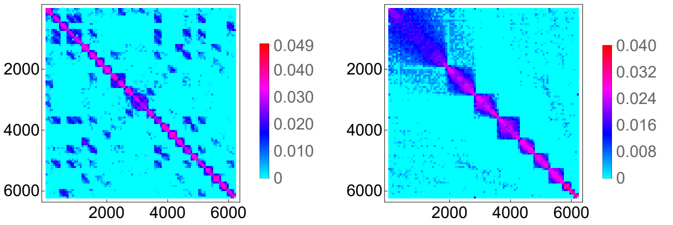
Fig. 7. The affinity matrices $\boldsymbol{W}$ for Smartphone dataset $(c^\star = 12)$.

References

[1] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering." *Nips*, vol. 17, no. 1601-1608, p. 16, 2004.

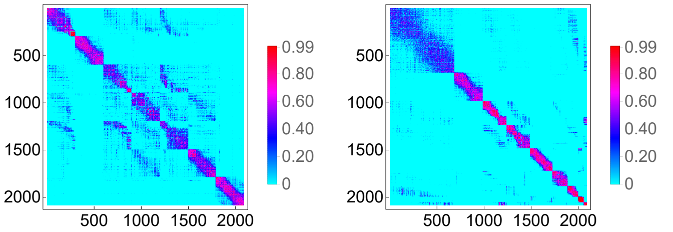
[2] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognit.*, vol. 41, no. 3, pp. 1012–1029, 2008.

[3] D. P. Hofmeyr, "Improving Spectral Clustering using the Asymptotic Value of the Normalised Cut," 2017.



(a) sorted by true clusters (b) sorted by estimated clusters

Fig. 8. The affinity matrices $\boldsymbol{W}$ for ISOLET dataset $(c^\star = 26)$.



(a) sorted by true clusters (b) sorted by estimated clusters

Fig. 9. The affinity matrices $\boldsymbol{W}$ for Images seg. dataset $(c^\star = 7)$.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[5] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst. 14*, pp. 849–856, 2001.

[6] U. von Luxburg, "A Tutorial on Spectral Clustering," *Stat. Comput.*, vol. 17, no. March, pp. 395–416, 2006.

[7] E. Forgy, "Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification," *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.

[8] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Appl. Stat.*, vol. 28, no. 1, p. 100, 1979.

[9] L. Mackey, "Deflation Methods for Sparse PCA." *Adv. Neural Inf. Process. Syst.*, pp. 1017–1024, 2008.

[10] A. Strehl and J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2002.

[11] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison," *Proc. 26th Annu. Int. Conf. Mach. Learn. - ICML '09*, vol. 11, pp. 1–8, 2009.

[12] G. W. G. W. Stewart and J.-g. Sun, *Matrix perturbation theory*. Academic Press, 1990.

[13] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis," *J. Comput. Graph. Stat.*, vol. 15, no. 2, pp. 265–286, 2006.

[14] M. Journée and Y. Nesterov, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, 2010.

[15] J. Song, P. Babu, and D. P. Palomar, "Sparse Generalized Eigenvalue Problem Via Smooth Optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1627–1642, 2015.

[16] K. Benidis, Y. Sun, P. Babu, and D. P. Palomar, "Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation," in *IEEE Trans. Signal Process.*, vol. 64, no. 23, 2016, pp. 6211–6226.

[17] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 63, no. 2, pp. 411–423, may 2001.

[18] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies," in *Data Min. Knowl. Discov.*, vol. 27, no. 3. Springer US, nov 2013, pp. 344–371.

[19] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Am. Stat. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Kdd*, vol. 96, no. 34, pp. 226–231, 1996.