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 indices of the nonzero components, then 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 \( S_p^n \) denote the set of \( n \)-dimensional symmetric positive semi-definite matrices. Define \( \mathbb{T}_n := \{1, 2, \ldots, n\} \). Given \( \mathcal{A} \subseteq \mathbb{T}_n \), define the indicator vector \( 1_{\mathcal{A}} \in \mathbb{R}^n \) by

\[
\forall i \in \mathbb{T}_n: \quad (1_{\mathcal{A}})_i := \begin{cases} 1 & i \in \mathcal{A} \\ 0 & i \notin \mathcal{A}. \end{cases}
\] (1)

Define \( 1 := 1_{\mathbb{T}_n} \). Given \( a \in \mathbb{R}^n \), let \( \text{diag}(a) \) denote the diagonal matrix with entries given by \( a \). Define the support \( \text{supp}(a) := \{ i \in \mathbb{T}_n | a_i \neq 0 \} \). Define the \( \ell_0 \) pseudo-norm \( \| a \|_0 := |\text{supp}(a)| \). The term \( i \)-th principal (minor) eigenvector refers to the eigenvector associated with the eigenvalue \( \lambda_i \) 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 \( \{x_1, \ldots, x_n\} \) and the corresponding affinity matrix \( W \in \mathbb{R}^n \times n \) whose entry \( w_{ij} = w_{ji} \) represents the affinity between \( x_i \) and \( x_j \). Consider a partition \( \{C_1, \ldots, C_c\} \) of the indices \( \mathbb{T}_n \) as clusters of the dataset. Let \( \sim \) denote the equivalence relation with respect to \( \{C_1, \ldots, C_c\} \), i.e.

\[
i \sim j \iff (\exists k \in \mathbb{T}_c \text{ s.t. } i, j \in C_k).\] (2)

Spectral clustering then seeks the clusters \( \{C_1, \ldots, C_c\} \) such that:

- intra-cluster affinity \( (w_{ij} \text{ for } i \sim j) \) is large.
- inter-clusters affinity \( (w_{ij} \text{ for } i \sim j) \) is small.

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

\[
L := D - W \quad (3)
\]
\[
L_{\text{sym}} := I - D^{-1/2}WD^{-1/2} \quad (4)
\]
\[
L_{\text{rw}} := I - D^{-1}W \quad (5)
\]

This work was supported in part by JSPS Grants-in-Aids (15K06081, 15K1396, 15H02757).
where $D := \text{diag}(W1)$. Since $W$ is symmetric and non-negative, the followings hold:

**Proposition 1** (Properties of the graph Laplacians [6]). (a) All eigenvalues of $L$, $L_{\text{sym}}$ and $L_{rw}$ are non-negative. (b) $L_{\text{sym}}v = \lambda v \iff L_{rw}(D^{-1/2}v) = \lambda (D^{-1/2}v)$ (c) Let $G$ be a graph whose weighted adjacency matrix is $W$, and $C_i \subseteq \{1, \ldots, m\}$ be the connected component of $G_i$. Then $L$ and $L_{rw}$ share the same eigenspace span $\{1_{c_1}, \ldots, 1_{c_c}\}$ corresponding to the eigenvalue 0.

We now consider the ideal case where the desired clusters $\{C_1, \ldots, C_c\}$ have exactly zero inter-cluster affinity:

$$\forall i, j \in \{1, \ldots, m\}, \quad i \sim j \iff w_{ij} = 0.$$  

(6)

The clusters $\{C_1, \ldots, C_c\}$ then coincide with the connected components $\{\tilde{C}_1, \ldots, \tilde{C}_m\}$ in Proposition 1. Therefore the minor eigenvectors $\{u_1, \ldots, u_c\}$ of $L$ give a basis of span $\{1_{c_1}, \ldots, 1_{c_c}\}$. Define $U := \begin{bmatrix} u_1 & \ldots & u_c \end{bmatrix} \in \mathbb{R}^{n \times c}$ and let $y_i \in \mathbb{R}^c$ be the $i$-th column of $U^\top$. The following property is useful to identify $\{C_1, \ldots, C_c\}$:

$$\forall i, j \in \{1, \ldots, n\}, \quad i \sim j \iff y_i = y_j.$$  

(7)

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

$$L = \hat{L} + E,$$  

(8)

then the minor eigenvectors of $\hat{L}$ form a basis of span $\{1_{c_1}, \ldots, 1_{c_c}\}$ as in the ideal case mentioned above. Furthermore, since $E$ has a small Frobenius norm, the eigenvectors of $L$ are close to those of $\hat{L}$ in a certain sense, according to the matrix perturbation theory [12]. This implies that (7) approximately holds, and the clusters $\{C_1, \ldots, C_c\}$ can be identified by clustering $\{y_1, \ldots, y_n\}$ 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 $\{u_1, \ldots, u_c\}$ can be an arbitrary orthonormal basis of span $\{1_{c_1}, \ldots, 1_{c_c}\}$, and there is no guarantee that it coincides with $\{1_{c_1}, \ldots, 1_{c_c}\}$. This complicates the subsequent clustering process, and thus one cannot compute $C_i$ 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 $\{u_1, \ldots, u_c\}$ of the graph Laplacian $L$. The regularization leads to obtaining the sparsest vectors, i.e. $\{1_{c_1}, \ldots, 1_{c_c}\}$, in the eigenspace span $\{1_{c_1}, \ldots, 1_{c_c}\}$. Then each cluster $C_i$ can be identified independently by taking the support of $\hat{u}_i$, 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 $A \in \mathbb{S}^n$, a first principal eigenvector $u(A)$ is given by

$$u(A) \in \arg\max_{u \in S^n} u^\top Au,$$  

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

$$\hat{u}(A) \in \arg\max_{u \in S^n} (u^\top Ax - \rho \|u\|_1),$$  

where $\rho > 0$ is the sparsity-controlling parameter.

Several feasible algorithms have been proposed to approximate $\hat{u}(A)$ for $A \in \mathbb{S}^n_+$, mainly in the context of sparse principal component analysis (SPCA) [13], [14], [15], [16].

A first minor sparse eigenvector $\hat{u}_1$ of $L$, which is needed for the proposed algorithm, is defined as $\hat{u}_1 := \hat{u}(-L) = \hat{u}(L_{\text{rev}})$, where $L_{\text{rev}} := \lambda_{\text{max}} I - L \in \mathbb{S}_+^n$ and $\lambda_{\text{max}}$ is the largest eigenvalue of $L$. The subsequent minor sparse eigenvectors $\{\hat{u}_2, \ldots, \hat{u}_c\}$ 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 $L$ tends to give poor results. Using normalized Laplacian $L_{\text{sym}}$ or $L_{\text{rev}}$ is therefore suggested in the proposed algorithm. Furthermore, $L_{\text{sym}}$ is better since Proposition 1 (c), which does not apply to $L_{\text{sym}}$, is the key to our derivation. The sparse eigenvectors of $L_{\text{sym}}$ however cannot be computed directly due to its asymmetry. Hence the proposed algorithm computes a minor sparse eigenvector $v_1$ of $L_{\text{sym}}$, and then regard $D^{-1/2}v_1$ as a sparse eigenvector of $L_{\text{rev}}$.

2) **Effects of the noise:** In the practical settings where the inter-cluster affinity is not exactly zero, $\{u_1, \ldots, u_c\}$ are not exactly $\{1_{c_1}, \ldots, 1_{c_c}\}$. This causes the following two issues:

(a) $\text{supp}(\hat{u}_i)$ is a good estimate of $C_i$ since $\hat{u}_i$ is likely to have very small entries such as $10^{-5}$.

(b) The resulting clusters may overlap.
To deal with issue (a), we use the following function instead of \( \text{supp}'_i \):
\[
\text{supp}'_i: \mathbf{x} \mapsto \left\{ i \in \overline{1,n}: |x_i| > r \max_{j \in \overline{1,n}} |x_j| \right\},
\]
where \( 0 < r < 1 \) is a parameter. As a remedy for issue (b), \( C_i \) is estimated by
\[
C_i := R_{i-1} \cap \text{supp}'_i (\hat{\mathbf{u}}_i),
\]
where \( R_i := R_{i-1} \setminus C_i \left( R_0 := \overline{1,n} \right) \) 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 \( \rho \) is too large. The optimization problem (11) is often relaxed using the \( \ell_1 \)-norm for tractability. Observe the following equation:
\[
\max_{u \in S^n} \|u\|_1 = \sqrt{n}.
\]
This implies that the “magnitude” of the \( \ell_1 \)-term depends on the number \( n \) of data points, and that \( \rho' := \sqrt{n} \rho \) 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 \( \text{supp}'_i \). The fixed \( r = 0.1 \) empirically gives good performances.

The proposed method is summarized in Algorithm 1.

Algorithm 1: Sparse Spectral Clustering (proposed)

1: Input: The affinity matrix \( \mathbf{W} \in \mathbb{R}^2 \times n \)
2: Parameter: \( \rho' > 0 \)
3: \( r \leftarrow 0.1 \) and \( \rho \leftarrow \rho' / \sqrt{n} \)
4: \( \mathbf{D} \leftarrow \text{diag}(\mathbf{W} \mathbf{1}) \)
5: \( \mathbf{L}_{\text{sym}} \leftarrow \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \)
6: \( \lambda_{\text{max}} \leftarrow (\text{the largest eigenvalue of } \mathbf{L}_{\text{sym}}) \)
7: \( \mathbf{A}_0 \leftarrow \lambda_{\text{max}} \mathbf{I} - \mathbf{L}_{\text{sym}} \)
8: \( i \leftarrow 0 \) and \( R_0 := \overline{1,n} \)
9: repeat
10: \( \mathbf{v}_i \leftarrow \arg\max_{u \in S^n} (\mathbf{v}^\top \mathbf{A}_i \mathbf{v} - \rho \|\mathbf{v}\|_0) \) (approximately)
11: \( \mathbf{A}_{i+1} \leftarrow \mathbf{A}_i - (\mathbf{v}_i^\top \mathbf{A}_i \mathbf{v}_i) \mathbf{v}_i \mathbf{v}_i^\top \)
12: \( \hat{\mathbf{u}}_i \leftarrow \mathbf{D}^{-1/2} \mathbf{v}_i \)
13: \( C_i := R_{i-1} \cap \text{supp}'_i (\hat{\mathbf{u}}_i) \)
14: \( R_i := R_{i-1} \setminus C_i \)
15: \( i \leftarrow i + 1 \)
16: until \( C_i = \emptyset \)
17: Output: \( C_1, \ldots, C_{i-1} \) and \( R_i \)

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 \( \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^2 \) 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 \( \mathbf{I} \). 150 points are i.i.d. sampled from each Gaussian, resulting \( n = 600 \) points in total. The affinity matrix \( \mathbf{W} \) is then set to the Gaussian kernel matrix \( \mathbf{K} \):
\[
k_{i,j} := \exp \left( -\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2\sigma^2} \right)
\]
with \( \sigma = 1 \).

The second dataset\(^1\) consists of three concentric circles with 62, 99, 138 data points, respectively \( (n = 299) \). For this data, \( \mathbf{W} \) is designed as follows:
\[
\mathbf{W} := \mathbf{K} \odot \mathbf{A},
\]
\(^1\)Available at www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

![Fig. 3. The affinity matrix \( \mathbf{W} \).](biconv.png)
results are measured based on normalized mutual information (NMI). NMI between two random variables \(X, Y\) is defined as follows:

\[
\text{NMI}(X, Y) := \frac{I(X, Y)}{\sqrt{H(X)H(Y)}},
\]

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

\[
H(X) := \mathbb{E}_X[-\log_2 p(X)]
\]

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

In our case \(X, Y\) correspond to the true label and the estimated label for each data point. Given true clusters \(\{C_1, \ldots, C_c\}\) and estimates \(\{\mathcal{C}_1, \ldots, \mathcal{C}_c\}\), the empirical joint probabilities are defined as

\[
\forall x \in \Gamma, c^*, y \in \Gamma, c: \quad p(x, y) := |\mathcal{C}_c^* \cap \mathcal{C}_y| / n.
\]

The following hold for all random variables \(X, Y\):

- \(0 \leq \text{NMI}(X, Y) \leq 1\);
- \(\text{NMI}(X, Y) = 0 \iff X, Y\) are independent;
- \(\text{NMI}(X, Y) = 1 \iff \text{knowing } X \text{ determines } Y, \text{ 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 \(c^*\) of clusters, the estimated number \(c\), and NMI are summarized in Table II. The datasets with NMI less than 0.7 are further investigated. Figure 6–9 illustrate the matrix \(W\) for these datasets. Figure 6–9 illustrate the matrix \(W\) for these datasets. The indices of \(W\) in each subfigure are sorted according to the true clusters or the clusters estimated by the proposed algorithm, respectively. Figure 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 \(W\).

### 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 \(\sigma\)). 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 \(\sigma, k\) for constructing \(W\), and the sparsity-inducing parameter \(\rho\) used in Algorithm 1.

<table>
<thead>
<tr>
<th>dataset</th>
<th>(n)</th>
<th>(d)</th>
<th>(\sigma)</th>
<th>(k)</th>
<th>(\rho)</th>
<th>standardized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt. Digits</td>
<td>5620</td>
<td>64</td>
<td>100</td>
<td>5</td>
<td>0.10</td>
<td>No</td>
</tr>
<tr>
<td>Pen Digits</td>
<td>10992</td>
<td>16</td>
<td>50</td>
<td>5</td>
<td>0.10</td>
<td>No</td>
</tr>
<tr>
<td>M.F. Digits</td>
<td>2000</td>
<td>216</td>
<td>10</td>
<td>5</td>
<td>0.10</td>
<td>Yes</td>
</tr>
<tr>
<td>Satellite</td>
<td>6435</td>
<td>36</td>
<td>30</td>
<td>5</td>
<td>0.10</td>
<td>No</td>
</tr>
<tr>
<td>Yale Faces</td>
<td>5850</td>
<td>1200</td>
<td>1000</td>
<td>5</td>
<td>0.10</td>
<td>No</td>
</tr>
<tr>
<td>Phoneme</td>
<td>4509</td>
<td>256</td>
<td>50</td>
<td>5</td>
<td>0.10</td>
<td>No</td>
</tr>
<tr>
<td>Smartphone</td>
<td>10929</td>
<td>561</td>
<td>30</td>
<td>5</td>
<td>0.06</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOLET</td>
<td>6238</td>
<td>617</td>
<td>30</td>
<td>5</td>
<td>0.10</td>
<td>Yes</td>
</tr>
<tr>
<td>Image Seg.</td>
<td>2310</td>
<td>19</td>
<td>50</td>
<td>50</td>
<td>0.10</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The similarity between the true labels and the clustering
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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm</th>
<th>Proposed</th>
<th>SPUDS</th>
<th>STSC</th>
<th>SCCD</th>
<th>Alt.SC</th>
<th>k-means</th>
<th>OCE</th>
<th>GMM</th>
<th>DBSCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>NMI</td>
<td>0.90</td>
<td>0.85</td>
<td>0.87</td>
<td>0.88</td>
<td>0.84</td>
<td>0.77</td>
<td>0.75</td>
<td>0.73</td>
<td>0.72</td>
</tr>
<tr>
<td>MNIST</td>
<td>NMI</td>
<td>0.94</td>
<td>0.92</td>
<td>0.92</td>
<td>0.91</td>
<td>0.90</td>
<td>0.88</td>
<td>0.86</td>
<td>0.83</td>
<td>0.80</td>
</tr>
<tr>
<td>Yale Faces</td>
<td>NMI</td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>PolyU</td>
<td>NMI</td>
<td>0.87</td>
<td>0.86</td>
<td>0.86</td>
<td>0.84</td>
<td>0.84</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.79</td>
</tr>
<tr>
<td>Phoneme</td>
<td>NMI</td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>Yale Faces</td>
<td>NMI</td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>ISOLET</td>
<td>NMI</td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>Satellite</td>
<td>NMI</td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>Average NMI</td>
<td></td>
<td>0.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.81</td>
<td>0.81</td>
<td>0.77</td>
</tr>
</tbody>
</table>

REFERENCES