

LSEC: Large-scale spectral ensemble clustering

Hongmin Li, Xiucui Ye, Akira Imakura and Tetsuya Sakurai

*Department of Computer Science
University of Tsukuba
Tsukuba, Japan*

li.hongmin.xa@alumni.tsukuba.ac.jp, {yexiucui, imakura,
sakurai}@cs.tsukuba.ac.jp

Abstract

Ensemble clustering is a fundamental problem in the machine learning field, combining multiple base clusterings into a better clustering result. However, most of the existing methods are unsuitable for large-scale ensemble clustering tasks due to the efficiency bottleneck. In this paper, we propose a large-scale spectral ensemble clustering (LSEC) method to strike a good balance between efficiency and effectiveness. In LSEC, a large-scale spectral clustering based efficient ensemble generation framework is designed to generate various base clusterings within a low computational complexity. Then all based clusterings are combined through a bipartite graph partition based consensus function into a better consensus clustering result. The LSEC method achieves a lower computational complexity than most existing ensemble clustering methods. Experiments conducted on ten large-scale datasets show the efficiency and effectiveness of the LSEC method. The MATLAB code of the proposed method and experimental datasets are available at <https://github.com/Li-Hongmin/MyPaperWithCode>.

1 Introduction

Ensemble clustering, also known as consensus clustering, is a classic problem in machine learning field, aiming to combine multiple base clustering into a better and more consensus clustering [24, 18, 6, 14, 28, 22, 32, 11, 12, 21, 20, 8, 10, 37, 38]. Due to its good performance, ensemble clustering has a pivotal role in many research areas, such as community detection [25] and bioinformatics [15, 29].

There are two critical steps in ensemble clustering: ensemble generation and consensus function. Ensemble generation aims to generate multiple base clusterings on the same datasets. In the early stage, k -means based ensemble generation methods [14, 27, 20] are widely used. Recently, spectral clustering based ensemble generation methods [13, 16] have received attention for its high

performance. On the other hand, the consensus function is used to integrate multiple base clusterings into a consensus one. We can roughly categorize ensemble clustering according to the consensus function into two categories: the co-association matrix based methods and the graph partitioning based methods.

The co-association matrix based ensemble clustering method [6, 14, 31, 30] is one of the most widely used ensemble clustering strategies. A typical example is the evidence accumulation clustering method [6], which counts the frequency of the pair-wise co-occurrence of the same cluster between a pair data points according to base clusterings. After treating the co-association matrix as a similarity matrix, the hierarchical agglomerative clustering algorithm is applied to obtain the consensus clustering. Iam-On et al. [14] extend the EAC method by constructing the co-association matrix based on the common neighborhood information between clusters. Tao et al. [26] propose a robust spectral ensemble clustering method to learn a robust representation for the co-association matrix by capturing the noises and conduct spectral clustering to obtain consensus clustering. Huang et al. [12] also enhance the co-association matrix based on similarity mapping from the cluster-level to the object-level and achieve ensemble clustering via fast propagation of cluster-wise similarities. However, the co-occurrence matrix based methods often lead to high computational cost, which has become a bottleneck for large-scale clustering tasks. Therefore, most co-association matrix based methods can work well in small-scale datasets but hardly complete large-scale clustering tasks in an acceptable time.

Graph partitioning based ensemble clustering methods [24, 5, 9, 16] aim to transform the ensemble clustering problem into a graph partitioning problem to find the consensus clustering. Strehl and Ghosh [24] construct a hypergraph representation by exploring base clusterings and propose three graph partitioning based ensemble clustering methods. Huang et al. [9] develop a sparse graph with a small number of probably reliable links from base clusterings and find the consensus clustering based on probability trajectory analysis. Li et al. [16] apply spectral clustering method as base clusterings and take the graph Laplacian matrices of base clusterings as input, then learn a consensus representation by optimizing the graph Laplacians of consensus clustering and base clusterings simultaneously, finally conduct spectral clustering to obtain consensus clustering. Although graph partitioning based methods have successfully improved clustering quality, they still have limitations regarding large-scale datasets.

Recently, a few studies have made progress in the application of large-scale data for ensemble clustering. Wu et al. [32] propose a k -means based consensus clustering (KCC) method, which applies the k -means method on a contingency matrix from base clusterings to obtain the consensus clustering result efficiently. Liu et al. [20] transform the spectral clustering of the co-association matrix into a weighted k -means method and prove two approaches are equivalent, which achieve high efficiency for ensemble spectral clustering. Huang et al. [13] point out the efficient bottleneck of k -means based ensemble generation and apply a large-scale spectral clustering method to fast product the base clusterings, then conduct bipartite graph partitioning to obtain the consensus clustering. Although these studies have achieved success in their respective

fields, the large-scale ensemble clustering problem is still a significant challenge due to its high computational complexity. Moreover, it is noteworthy that the ensemble generation step considerably takes up the run-time during large-scale ensemble clustering tasks, which has been rarely investigated in the literature.

In light of this, we propose a large-scale ensemble spectral clustering (LSEC) method to alleviate the problem of the application of ensemble clustering for large-scale data. In LSEC, a spectral clustering based ensemble generation method is designed to handle nonlinear datasets efficiently and provide high-quality base clusterings. The ensemble generation process is further accelerated by reusing K -nearest neighbors among base clusterings and using light- k -means to obtain the clustering results. After ensemble generation, a bipartite graph between data points and clusters from base clusterings is constructed to produce consensus clustering through the bipartite graph partitioning method efficiently. Experimental results on ten large-scale data sets demonstrate that LSEC delivers highly efficient and high-quality clustering performance compared to some state-of-the-art consensus clustering methods.

The main contributions of the proposed method are summaries as follows:

- An efficient spectral clustering based ensemble generation method is designed to handle large-scale datasets and provide high-quality base clusterings via divide-and-conquer based large-scale spectral clustering method.
- Two accelerating tricks are proposed: 1) the computation of similarity among multiple base clusterings is accelerated by reusing the K -nearest neighbors; 2) the process of obtaining base clustering results is accelerated by the light- k -means method.
- The proposed method efficiently generates base clusterings and conducts bipartite graph partitioning to find the consensus clustering. Its computational and space complexity is dominated by $O(\frac{m}{q}N\alpha d)$ and $O(NK)$, which achieves a lower computational complexity than most existing ensemble clustering methods.

2 Preliminaries

2.1 Ensemble Clustering

Ensemble clustering aims to combine multiple base clustering algorithms to achieve better clustering results. Let X be a dataset $X = \{x_1, \dots, x_n\}$ with n data points. The ensemble generation is the first step, which applies a specific clustering algorithm to produce m base clusterings. Let $\Pi = \{\pi_1, \dots, \pi_m\}$ be a set of base clusterings, where π_i is i -th base clustering and $\pi_i = \{\pi_i(x_1), \pi_i(x_2), \dots, \pi_i(x_n)\}$ indicates the clustering labels for all data points. Many studies [6, 14, 32, 19, 26, 9] use k -means based ensemble generation while some studies [13, 16] points out that spectral clustering based ensemble generation can significantly improve clustering quality on the nonlinear datasets. After ensemble generation, the

consensus function is used to integrate all base clusterings into a consensus one, which is the second step.

2.2 Divide-and-conquer based large-scale spectral clustering algorithm

Divide-and-conquer based large-scale spectral clustering algorithm (DnC-SC) has been proposed as an effective method for large-scale clustering tasks [17]. It first constructs an approximate similarity matrix via a divided-and-conquer based landmark selection and approximates K -nearest landmark searching. Then, it transfers the original spectral clustering problem into a bipartite graph partition problem to find the low-dimensional embedding by solving a smaller eigenproblem. Finally, it applies k -means on the low-dimensional embedding to obtain the final clustering result.

Let $R = \{r_1, r_2, \dots, r_p\}$ denote a set of landmarks, where $r_i \in \mathbb{R}^d$ has the same dimension as x_i . The divided-and-conquer based landmark selection is designed to generate a set of landmark points which can best represent the original data X . The objective function (1) measure how well R represent X by compute the residual sum of squares (RRS) between each x_j and its nearest r_i .

$$\zeta = \sum_{i=1}^p \sum_{x_j \in S_i} \|x_j - r_i\|^2, \quad (1)$$

where ζ denotes RSS and S_1, S_2, \dots, S_p indicate the subsets that are nearest to r_1, r_2, \dots, r_p , respectively. For each subset S_i , r_i is the subset center. The objective function (1) can be rewritten as follows:

$$g(X, p) = \arg \min_{S_1, \dots, S_p} \sum_{i=1}^p \sum_{x_j \in S_i} \|x_j - r_i\|^2. \quad (2)$$

The recursive functions (3) and (4) are used to divide the optimization problem into small sub-problems which are easier to be solved. The parameter α is used to determine the upper bound of k_i , which controls the landmark selection rate.

$$g(Q, h) = \bigcup_{i=1}^m g(A_i, k_i), \quad (3)$$

$$\{A_1, \dots, A_m\} = g(Q, m), \quad (4)$$

The light- k -means algorithm [17] is used to solve the larger dividing process $g(\cdot)$ (with more than $10p$ samples), which randomly selects a part of samples to find subset by k -means and then assign remained data points to the nearest subsets. For the smaller dividing processes (with less than or equal to $10p$ samples), k -means is directly used to find the subsets.

The similarities between each $x_i \in X$ and its K -nearest landmarks are used to construct a sparse similarity matrix. The centers' nature of landmarks is

used to estimate the K -nearest landmarks. Let S_{x_i} be the subset and $x_i \in S_{x_i}$. Denote $r_{x_i}^1$ is the landmark that is the center of S_{x_i} . According to the center's nature of landmark, $r_{x_i}^1$ is treated as the nearest landmark of x_i . In DnC-SC, a set of K' -nearest landmarks ($K' > K$) of $r_{x_i}^1$ is first obtained, denoted as $N_{K'}(r_{x_i}^1)$; then K -nearest landmark of x_i are searched from $N_{K'}(r_{x_i}^1)$, denoted as $N_K(x_i)$. Finally, the sparse similarity matrix B is constructed as follows [35, 36]:

$$b_{ij} = \begin{cases} \exp\left(\frac{-\|x_i - r_j\|^2}{2\sigma^2}\right), & \text{if } r_j \in N_K(x_i), \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where the Gaussian kernel is used to measure the similarity and σ is the bandwidth parameter.

The similarity matrix B reflects the relationship between data X and landmarks R , which can be treated as the edge of the bipartite graph $G(X, R, B)$. Therefore, the spectral clustering problem is converted into a bipartite graph partition problem. According to [17], the low-dimensional embedding of R side can be computed as follows:

$$L_R V = \lambda D_R V, \quad (6)$$

$$U = D_X^{-1} B V. \quad (7)$$

where $L_R = D_R - B^T D_X^{-1} B$, $D_X \in \mathbb{R}^{n \times n}$ and $D_R \in \mathbb{R}^{p \times p}$ are the diagonal matrices whose entries are $d_X(i, i) = \sum_{j=1}^n B_{ij}$ and $d_R(j, j) = \sum_{i=1}^n B_{ij}$, respectively. (6) is a small eigen-problem with size $p \times p$. U is the c bottom eigenvectors of X side. Finally, k -means is conducted on U to find c clusters as the final clustering result.

3 Proposed Framework

To improve the scalability of ensemble clustering, we propose the LSEC method that complies with the large-scale spectral clustering based formulation and aims to break through the efficiency bottleneck of previous algorithms. LSEC method consists of two steps: (1) Large-scale spectral clustering based ensemble generation: we design a new framework that applies the state-of-the-art large-scale spectral clustering algorithm to product base clusterings and further accelerate the process by reusing the K -nearest landmarks and using light- k -means to obtain base clustering results. (2) Bipartite graph partitioning based consensus function: we construct a bipartite graph between data points and clusters from base clusterings and obtain the consensus clustering result by bipartite graph partitioning. Fig. 1 shows an overview of proposed method.

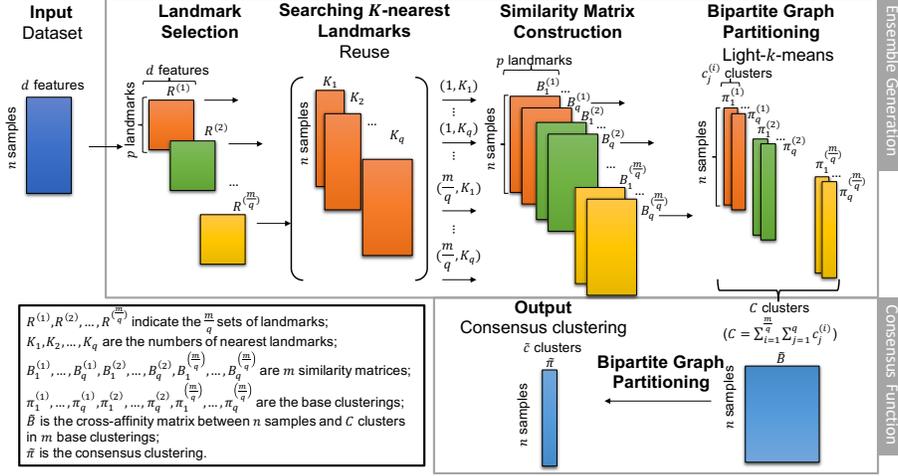


Figure 1: An overview of proposed method. Given a dataset, $\frac{m}{q}$ sets of landmarks are first generated, then a set of K -nearest neighbors are found for each $R^{(i)}$ and m sparse similarity matrices are constructed, finally the base clusterings are obtained through a bipartite graph partitioning process. The proposed method accelerates the similarity matrix construction by recycling K -nearest neighbors and bipartite graph partitioning by applying light- k -means.

3.1 Ensemble Generation based on Large-scale Spectral Clustering

The ensemble generation step aims to produce diverse m base clusterings with high efficiency. To improve the scalability of ensemble generation, we consider the divide-and-conquer based large-scale spectral clustering [17] as the base clustering algorithm, which can better handle nonlinear datasets than transitional clustering algorithm like k -means and maintain high efficiency. For better diversity of base clusterings and higher efficiency, we construct similarity matrices with multiple K -nearest neighbors graph of sparsification via reusing the K -nearest landmarks. Moreover, the bipartite graph partitioning is accelerated by applying light- k -means to obtain the clustering results.

3.1.1 Landmark Selection

First, the $\frac{m}{q}$ sets of landmarks are independently generated by solving the optimization problem (2). We recursively apply (3) and (4) to find an approximate local solution and turn the subset centers as landmarks. Let $R^{(i)} = \{r_1^{(i)}, r_2^{(i)}, \dots, r_p^{(i)}\}$ is a set of landmarks. Repeat the divide-and-conquer based landmark selection $\frac{m}{q}$ times, we have $\frac{m}{q}$ sets of landmarks as follows:

$$\mathcal{R} = \{R^{(1)}, R^{(2)}, \dots, R^{(\frac{m}{q})}\}, \quad (8)$$

where $R^{(i)}$ indicates the i -th set of landmarks and R is a set containing all $R^{(i)}$, $i = 1, 2, \dots, \frac{m}{q}$. The generation of each $R^{(i)}$ costs $O(N\alpha d)$ time complexity and constructing \mathcal{R} totally costs $O(\frac{m}{q}N\alpha d)$ time complexity.

3.1.2 Searching K -nearest landmarks

To construct a sparse similarity matrix with K -nearest neighbor sparsification, we need to search the K -nearest landmarks for each data point x_i . We consider constructing multiple similarity matrices with different sparsification of K -nearest landmarks for better diversity of base clusterings. Let $K_1 < K_2 < \dots < K_q$ be a set of numbers. We search K_1, K_2, \dots, K_q -nearest landmarks for each x_i , denoted as $N_{K_1}(x_i), N_{K_2}(x_i), \dots, N_{K_q}(x_i)$. According to the definition of K -nearest neighbors, we have

$$N_{K_1}(x_i) \subset N_{K_2}(x_i) \subset \dots \subset N_{K_q}(x_i). \quad (9)$$

That is, $N_{K_{j_1}}(x_i)$ is a subset of $N_{K_{j_2}}(x_i)$ if $K_{j_1} < K_{j_2}$. Therefore, we only need to compute K_q -nearest landmarks and then obtain the other K_1, K_2, \dots, K_q -nearest landmarks based on it without recomputing. We call this process reusing the nearest landmarks. Reusing the nearest landmarks accelerates the process of spectral clustering based ensemble generation. It directly reduces the computational time in two high-cost steps, landmark selection and searching K -nearest landmarks, by nearly q times. Besides efficiency, it also enhances the diversity of base clusterings by exploring multiple nearest neighbor graphs, which is helpful to improve the effectiveness of the proposed method.

3.1.3 Similarity Matrix Construction

Then, the sparse similarity matrix between X and each $R^{(i)}$ is constructed according to (5). Instead of constructing one similarity matrix for one set of landmarks, we build multiple similarity matrices using different sets of landmarks. For each $R^{(i)}$, we construct q sparse similarity matrices with K_1, K_2, \dots, K_q -nearest landmarks according to (5), respectively. We construct m similarity matrices as follows:

$$\mathcal{B} = \{B_1^{(1)}, \dots, B_q^{(1)}, B_1^{(2)}, \dots, B_q^{(2)}, \dots, B_1^{(\frac{m}{q})}, \dots, B_q^{(\frac{m}{q})}\}, \quad (10)$$

where $B_j^{(i)}$ indicates a similarity matrix between X and $R^{(i)}$ with sparsification of K_j -nearest landmarks, \mathcal{B} is a set containing all $B_j^{(i)}$ and the total size of \mathcal{B} is m . The computational cost to obtain a sparse similarity matrix $B^{(i)j}$ is $O(NK_j d)$ [17]. By reusing the nearest landmarks, we can generate \mathcal{B} with only $O(\frac{m}{q}NK_q d)$ computational cost. For convenience, we will use K instead of K_q to show the computational complexity in the rest of the paper.

3.1.4 Bipartite Graph Partitioning

After obtaining m similarity matrices, we treat each $B_j^{(i)}$ as the edge of a bipartite graph $G(X, R^{(\lceil \frac{i}{q} \rceil)}, B^{(i)j})$ and solve a bipartite graph partition problem by

(6) and (7) to construct a $c_j^{(i)}$ -dimensional embedding denoted as $U^{(i)j}$. Note that $c_j^{(i)}$ is also the number of clusters. It costs $O(p^3)$ time complexity to solve each bipartite graph partition problem (6) and $O(NK(K + c_j^{(i)}))$ to compute the each $c_j^{(i)}$ -dimensional embedding. The cluster number of $c_j^{(i)}$ is randomly selected as follow:

$$c_j^{(i)} = \lfloor \tau(c_{max} - c_{min}) \rfloor + c_{min}, \quad (11)$$

where $\tau \in [0, 1]$ is a random variable and c_{max} and c_{min} are the upper and lower bounds of the cluster number, respectively.

The obtained $c_j^{(i)}$ eigenvectors are stacked to form a new matrix, upon which the light- k -means [17] is applied to construct the base clustering result. In light- k -means, a set of p' samples are first randomly selected as representatives, then c clusters centers are generated by applying k -means clustering on p' representatives, finally, assign labels to remained samples according to their nearest cluster centers. The computational complexity of light- k -means is $O(pcdt + Ncd)$, where $O(Ncd)$ is the dominated term and d is the dimensional size. The light- k -means alleviates the computational cost from t iterations and can achieve more efficiency on the platform optimized for matrix operation. The use of light- k -means significantly accelerates the process of obtaining base clusterings for large-scale datasets. Finally, m base clusterings are generated, which are represented as

$$\Pi = \{\pi_1^{(1)}, \dots, \pi_q^{(1)}, \pi_1^{(2)}, \dots, \pi_q^{(2)}, \dots, \pi_1^{(\frac{m}{q})}, \dots, \pi_q^{(\frac{m}{q})}\}, \quad (12)$$

where $\pi_j^{(i)}$ denotes a base clustering with $c_j^{(i)}$ clusters. For convenience, we use c instead of $c_j^{(i)}$ to show the computational complexity in the rest paper. The computational complexity of using light- k -means is $O(Nc^2 + p'c^2t)$, where $O(Nc^2)$ is the dominated term. Overall, the computational complexity of the bipartite graph partition is $O(m(N(K^2 + c^2 + Kc) + p^3))$. We summarize the ensemble generation process of the proposed method in algorithm 1.

3.2 Consensus Function based on Bipartite Graph Partitioning

After ensemble generation, the base clusterings will be combined according to a consensus function for obtaining the consensus partition. Again, we treat this problem as a bipartite graph partition problem and give a similar solution like section 2.2.

To define the bipartite graph, we first collect all clusters though the base clusterings as (12) and we denotes the clusters in (13) for clarity.

$$\Psi = \{\Omega_1^{(1)}, \dots, \Omega_q^{(1)}, \Omega_1^{(2)}, \dots, \Omega_q^{(2)}, \dots, \Omega_1^{(\frac{m}{q})}, \dots, \Omega_q^{(\frac{m}{q})}\}, \quad (13)$$

where $\Omega_j^{(i)}$ indicate the set of clusters in $\pi_j^{(i)}$. There are $c_j^{(i)}$ clusters in each

Algorithm 1: Proposed ensemble generation

Input: Dataset X , number of base clusterings m , a set of number of K -nearest landmarks K_1, K_2, \dots, K_q

Output: base clusterings $\pi_1, \pi_2, \dots, \pi_m$

- 1 Solve (1) by recursively applying (3) and (4) to obtain $\frac{m}{q}$ sets of landmarks \mathcal{R} ;
- 2 **for** $i \leftarrow 1$ **to** $\frac{m}{q}$ **do**
- 3 Search K_q -nearest landmarks of each data points according to [17];
- 4 **for** $j \leftarrow 1$ **to** q **do**
- 5 Obtain K_j -nearest landmarks of each data points according to (9);
- 6 Construct similarity matrix between X and R^i with sparsification of K_j -nearest landmarks by (5);
- 7 **end**
- 8 **end**
- 9 Collect all similarity matrices \mathcal{B} by (10);
- 10 **for** $i \leftarrow 1$ **to** m **do**
- 11 Find a low-dimensional embedding U by (6) and (7);
- 12 Apply light- k -means on the embedding U to obtain base clustering π_i .
- 13 **end**

Table 1: The cluster indicator matrix

	ω_1	ω_2	\dots	ω_C	Σ
\mathcal{X}_1	\tilde{b}_{11}	\tilde{b}_{12}	\dots	\tilde{b}_{1C}	m
\mathcal{X}_2	\tilde{b}_{21}	\tilde{b}_{22}	\dots	\tilde{b}_{2C}	m
\cdot	\cdot	\cdot	\dots	\cdot	\cdot
\mathcal{X}_n	\tilde{b}_{n1}	\tilde{b}_{n2}	\dots	\tilde{b}_{nC}	m
Σ	$\ \omega_1\ $	$\ \omega_2\ $	\dots	$\ \omega_C\ $	Nm

$\Omega_j^{(i)}$, which we denote as:

$$\Omega_j^{(i)} = \{\omega'_1, \omega'_2, \dots, \omega'_{c_j^{(i)}}\}, \quad (14)$$

where ω'_t is the t -th cluster in $\Omega_j^{(i)}$. Thus, the total number of clusters in Ψ can be counted as $C = \sum_{i=1}^m \sum_{j=1}^q c_j^{(i)}$. For convenience, we simplify the notation of (15) as follows:

$$\Psi = \{\omega_1, \omega_2, \dots, \omega_C\}, \quad (15)$$

After the definition of Ω , we design a bipartite graph between data points and clusters as follow:

$$\tilde{G} = \{\mathcal{X}, \Omega, \tilde{B}\}, \quad (16)$$

where \tilde{B} is the cross-affinity matrix between \mathcal{X} and Ω . \tilde{B} can also be interpreted as the cluster indicator matrix of X . Table 1 shows the cluster indicator matrix, where $b_{ij} = 1$ indicates that $X_i \in \omega_j$. \tilde{G} is an unweighted bipartite graph where any edge between node X_i and ω_j indicates the cluster relationship $X_i \in \omega_j$. We can give the formula of \tilde{B} as follow:

$$\tilde{b}_{ij} = \begin{cases} 1, & \text{if } x_i \in \omega_j, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

As Table 1 shows, the sum of each rows of \tilde{B} is as the same as number of base clusterings m because there is only one cluster x_i belongs to each base clustering π^j , i.e., $\forall i' \neq j'$, if $\omega_{i'} \in \pi^i$ and $\omega_{j'} \in \pi^i$, then $\omega_{i'} \cap \omega_{j'} = \emptyset$. Though the number of samples in each ω_i is uncertain, i.e., $\|\omega_i\|$, the total number of non-zeros entries is clearly Nm (see Table 1).

For this modified bipartite graph \tilde{G} , we consider a similar partition strategy to what we introduced in Section 2.2. According to [17], we can write the full similarity of \mathcal{G} as follow

$$\tilde{W} = \begin{bmatrix} 0 & \tilde{B} \\ \tilde{B}^T & 0 \end{bmatrix}. \quad (18)$$

Then the we have a generalized eigen-problem of \tilde{G}

$$\tilde{L}\tilde{f} = \lambda\tilde{D}\tilde{f}, \quad (19)$$

where $\tilde{L} = \tilde{D} - \tilde{W}$ and \tilde{D} is a diagonal matrix with $\tilde{d}_{ii} = \sum_{j=1}^n \tilde{w}_{ij}$. According to (20) and (7), we design a smaller eigen-problem to compute the eigenvector \tilde{U} in \mathcal{X} side as follows:

$$L_\Omega \tilde{V} = \tilde{\lambda} D_\Omega \tilde{V}, \quad (20)$$

where $L_\Omega = \tilde{D}_\Omega - \tilde{B}^\top \tilde{D}_\mathcal{X}^{-1} \tilde{B}$ is the graph Laplacian, $\tilde{D}_\mathcal{X} \in R^{n \times n}$ and $\tilde{D}_\mathcal{R} \in R^{p \times p}$ are the diagonal matrices whose entries are $\tilde{d}_\mathcal{X}(i, i) = \sum_{j=1}^n \tilde{B}_{ij}$ and $\tilde{d}_\mathcal{R}(j, j) =$

$\sum_{i=1}^n \tilde{B}_{ij}$, respectively. The size of L_Ω is $C \times C$. Solving the eigen-problem (20) cost $O(C^3)$ computational time. Substituting \tilde{V} into (7), we can computer \tilde{U} as follow

$$\tilde{D} = \tilde{D}_X^{-1} \tilde{B} \tilde{V}. \quad (21)$$

The \tilde{c} bottom eigenvectors \tilde{U} can be computed with with $O(Nm(m+c))$ time. Finally, the consensus clustering results in LSEC can be obtained by the k -means method with $O(Nc^2t)$ time. We summarize the proposed method LSEC in algorithm 2.

Algorithm 2: Large-scale ensemble spectral clustering

Input: Dataset X , number of base clusterings m , a set of number of K -nearest landmarks K_1, K_2, \dots, K_q , m base clusterings, number of clusters \tilde{c}

Output: Consensus clustering $\tilde{\pi}$

- 1 Produce m base clustering by large-scale ensemble generation;
 - 2 Construct the cluster indicator matrix \tilde{B} according to (17);
 - 3 Solve the eigen-problem (20) to compute \tilde{V} ;
 - 4 Find a low-dimensional embedding \tilde{U} of X by (21);
 - 5 Applying k -means to find \tilde{c} clusters on \tilde{U} to obtain consensus clustering.
Obtain consensus clustering by large-scale consensus function.
-

4 Discussion

4.1 Computational Complexity Analysis

In this section, we summarize the computational cost of the proposed method. The ensemble generation of LSEC algorithm takes $O(mN(\alpha d + K^2 + Kc + Kd + qc^2) + p^3 + p^2(d + K))$ computational cost. The consensus function of LSEC takes $O(N((qm)^2 + qmk + c^2t) + C^3)$ time. With consideration to $m, q, k, K < \alpha \ll p \ll N$, the dominant term of the overall time complexity of LSEC is $O(Nm(\alpha d + qk^2))$. Meanwhile, the memory costs of the ensemble generation and the consensus function of our LSEC algorithm are respectively $O(N\alpha)$ and $O(Nm)$. Table 2 provides a comparison of the computational complexity of our DnC-SC algorithm against a state-of-the-art large-scale ensemble clustering method U-SPEC.

Table 2: Comparison of the computational complexity between LSEC and U-SPEC.

Method	Ensemble Generation			Consensus Function
	Landmark selection	Similarity construction	Bipartite graph partitioning	
U-SPEC	$O(mp^2dt)$	$O(mNp^2d)$	$O(m(N(K^2 + c^2t + Kc) + p^3))$	$O(N(m^2 + mk + c^2t) + C^3)$
LSEC	$O(\frac{m}{q}N\alpha d)$	$O(\frac{m}{q}NKd)$	$O(m(N(K^2 + c^2 + Kc) + p^3))$	$O(N(m^2 + mk + c^2t) + C^3)$

4.2 Relations with Other Methods

As a large-scale spectral ensemble clustering method, the proposed method is closely related to the U-SENC method in [13]. We compare the proposed method with the U-SENC to discuss the improvements of the proposed method.

Firstly, we compare them to the ensemble generation methods in the term of diversity and efficiency. In U-SENC, base clusterings are directly generated using a large-scale spectral clustering U-SPEC with different numbers of clusters. As a large-scale spectral clustering method, the U-SPEC method also uses the landmark selection technique. Thus, the diversity of base clusterings of U-SENC is from two facts: the different landmarks and the number of clusters of ensemble generation. However, the K -nearest neighbor graph is not used to improve the diversity in U-SENC further. In our proposed method, we consider the various landmarks and number of clusters and use different K -nearest neighbors to construct a sparse similarity matrix to improve the overall diversity of base clustering.

Secondly, we compare them to the ensemble generation methods in the term of efficiency. Since the different K -nearest neighbor graphs can share the same K -nearest neighbors between data points and landmarks, the computational complexity of similarity matrix construction is much less than the U-SENC method. For large-scale datasets, another computational bottleneck is the final k -means step of large-scale spectral clustering. In our proposed method, we use the light- k -means to accelerate the base clustering results, significantly improving large-scale datasets' efficiency.

Overall, LSEC redesigns the ensemble generation framework based on a more efficient clustering method (i.e., DnC-SC) and accelerates the process by reusing the K -nearest neighbors among multiple base clusterings. Furthermore, a light- k -means method is used to fast obtain the base clustering results. The computational complexity of the proposed method is faster than most existing large-scale ensemble clustering methods.

5 Experiments

In this section, we conduct experiments on five real and five synthetic datasets to evaluate the performance of the proposed LSEC method. The comparison experiments against several state-of-the-art spectral clustering methods show better clustering quality and efficiency for LSEC methods. Besides that, the analysis of parameters is performed. For each experiment, the test method is repeated 20 times, and the average performance is reported. All experiments are conducted in Matlab R2020a on a Mac Pro with 3 GHz 8-Core Intel Xeon E5 and 16 GB of RAM.

5.1 Datasets and Evaluation Measures

Our experiments are conducted on ten large-scale datasets, varying from nine thousands to as large as twenty million data points. Specifically, the five real

Table 3: Properties of the real and synthetic datasets.

Dataset	#Object	#Dimension	#Class	
<i>Real</i>	<i>USPS</i>	9298	256	10
	<i>PenDigits</i>	10,992	16	10
	<i>Letters</i>	20,000	16	26
	<i>MNIST</i>	70,000	784	10
	<i>Covertypes</i>	581,012	54	7
<i>Synthetic</i>	<i>TB-1M</i>	1,000,000	2	3
	<i>SF-2M</i>	2,000,000	2	4
	<i>CC-5M</i>	5,000,000	2	3
	<i>CG-10M</i>	10,000,000	2	11
	<i>FL-20M</i>	20,000,000	2	13
	<i>CG-10M</i>	10,000,000	2	11

datasets are *PenDigits* [1]¹, *USPS* [4]², *Letters* [7]³, *MNIST* [3], and *Covertypes* [2]⁴. The five synthetic datasets are *Two Bananas (TB-1M)*, *Smiling Face-2M (SF-2M)*, *Concentric Circles-5M (CC-5M)*, *Circles and Gaussians-10M (CG-10M)*, *Flower-20M (FL-20M)* [13]⁵. Fig. 2 shows the synthetic datasets. The properties of the datasets are summarized in Table 3.

We adopt two widely used evaluation metrics, i.e., Normalized Mutual Information (NMI) [23] and Accuracy (ACC) [34], to evaluate the clustering results. Let $X = [x_1, x_2, \dots, x_n]$ be the data matrix. For each data point x_i , denote $\pi_t(x_i)$ and $\pi_c(x_i)$ as the cluster label of ground truth and obtained cluster label from clustering methods, respectively. The ACC is defined as:

$$\text{ACC} = \frac{\sum_{i=1}^n \delta(\pi_t(x_i), \text{map}(\pi_c(x_i)))}{n}, \quad (22)$$

where n is the number of data and $\delta(\pi_t(x_i), \pi_c(x_i))$ is a function to check $\pi_t(x_i)$ and $\pi_c(x_i)$ are equal or not, returning 1 if equals otherwise returning 0. The $\text{map}(\pi_c(x_i))$ is a best mapping function that maps each predicted label to the most possibly true cluster label by permuting operations [33].

The NMI is the normalization of Mutual information by the joint entropy as follow:

$$\text{NMI} = \frac{\sum_{\pi_t(x_i) \in T, \pi_c(x_i) \in C} p(\pi_t(x_i), \pi_c(x_i)) \ln \frac{p(\pi_t(x_i), \pi_c(x_i))}{p(\pi_t(x_i))p(\pi_c(x_i))}}{-\sum_{\pi_t(x_i) \in T, \pi_c(x_i) \in C} p(\pi_t(x_i), \pi_c(x_i)) \ln(p(\pi_t(x_i), \pi_c(x_i)))}, \quad (23)$$

¹<https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits>

² <http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html>

³<https://archive.ics.uci.edu/ml/datasets/Letter+Recognition>

⁴<https://archive.ics.uci.edu/ml/datasets/covertypes>

⁵<https://www.researchgate.net/publication/330760669>

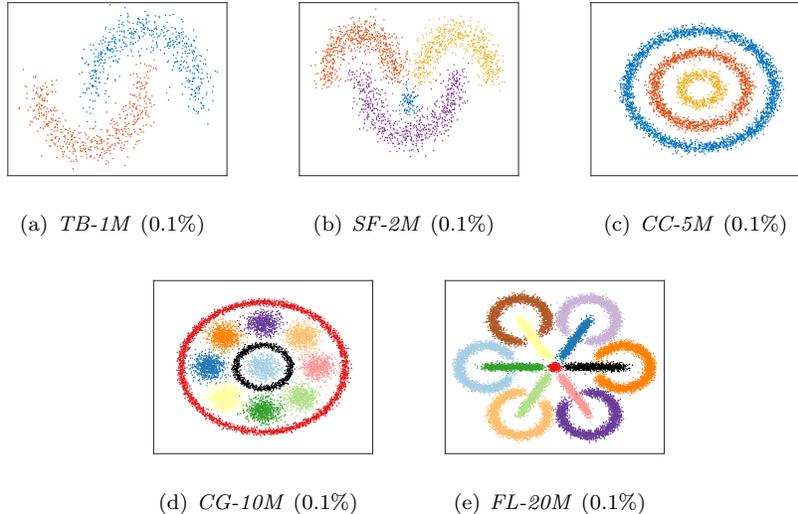


Figure 2: Illustration of the five synthetic datasets. Note that only 0.1% samples of each dataset are plotted.

A better clustering result will provide a larger value of NMI/ACC. Both NMI and ACC are in the range of $[0, 1]$.

5.2 Compared Methods and Experimental Settings

In this experiments, we compare the proposed method with one baseline clustering method, i.e., the divide-and-conquer based large-scale spectral clustering (DnC-SC) [17], as well as seven state-of-the-art large-scale spectral clustering methods. The compared spectral clustering methods are listed as follows:

1. **EAC** [6]: evidence accumulation clustering.
2. **KCC** [32]: k -means based consensus clustering.
3. **PTGP** [9]: probability trajectory based graph partitioning.
4. **SEC** [19]: spectral ensemble clustering.
5. **LWEA** [11]: locally weighted evidence accumulation.
6. **LWGP** [11]: locally weighted graph partitioning.
7. **U-SENC** [13]: ultra-scalable ensemble clustering.

There are several common parameters among the methods mentioned above. We set these parameters as follow:

Table 4: ACC(%) scores (over 20 runs) by our methods and the baseline ensemble clustering methods (The best score in each row is in bold).

Dataset	DnC-SC	EAC	KCC	PTGP	SEC	LWGP	U-SENC	LSEC
<i>PenDigits</i>	82.27 \pm 1.33	77.67 \pm 2.30	44.68 \pm 5.10	80.89 \pm 1.26	32.37 \pm 3.88	73.66 \pm 2.14	87.02 \pm 1.65	88.26 \pm 2.10
<i>USPS</i>	82.55 \pm 1.96	66.76 \pm 1.69	56.37 \pm 3.41	67.14 \pm 0.26	40.77 \pm 5.82	65.82 \pm 3.41	78.29 \pm 2.39	80.97 \pm 5.31
<i>Letters</i>	33.54 \pm 1.21	29.79 \pm 0.60	24.46 \pm 1.24	26.66 \pm 1.42	23.60 \pm 1.28	27.88 \pm 0.78	37.03 \pm 1.28	36.33 \pm 0.89
<i>MINST</i>	74.24 \pm 2.14	N/A	45.61 \pm 4.96	66.96 \pm 0.68	33.15 \pm 2.07	56.27 \pm 1.47	75.48 \pm 3.01	80.19 \pm 3.74
<i>Coverttype</i>	23.48 \pm 1.86	N/A	32.52 \pm 0.41	23.45 \pm 0.96	39.63 \pm 6.15	30.64 \pm 0.42	21.34 \pm 1.06	23.42 \pm 1.86
<i>TB-1M</i>	99.62 \pm 0.02	N/A	67.76 \pm 1.41	81.95 \pm 0.00	67.94 \pm 3.66	99.71 \pm 0.45	99.75 \pm 0.01	99.72 \pm 2.31
<i>SF-2M</i>	9.43 \pm 0.31	N/A	50.94 \pm 4.15	60.25 \pm 0.94	49.88 \pm 5.68	80.04 \pm 3.45	76.54 \pm 2.88	85.17 \pm 9.18
<i>CC-5M</i>	99.98 \pm 0.00	N/A	72.25 \pm 6.41	34.95 \pm 0.00	41.57 \pm 0.81	97.84 \pm 3.71	99.99 \pm 0.00	95.61 \pm 11.67
<i>CG-10M</i>	66.83 \pm 4.46	N/A	58.12 \pm 5.41	60.89 \pm 1.73	46.26 \pm 5.72	71.95 \pm 3.19	82.34 \pm 5.59	97.57 \pm 3.49
<i>FL-20M</i>	81.90 \pm 5.61	N/A	48.21 \pm 4.14	51.21 \pm 1.41	41.70 \pm 0.42	72.15 \pm 2.45	78.16 \pm 3.21	82.81 \pm 3.21
Avg. score	-	N/A	50.09	55.44	41.69	67.60	72.59	77.01
Avg. rank	-	6.00	5.00	4.00	5.60	3.30	2.30	1.90

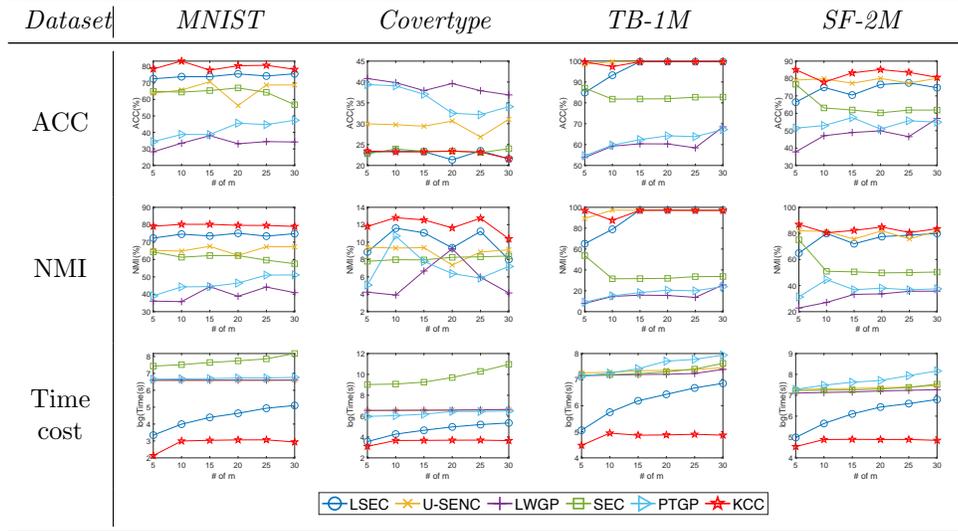
Table 5: NMI(%) scores (over 20 runs) by our methods and the baseline ensemble clustering methods (The best score in each row is in bold).

Dataset	DnC-SC	EAC	KCC	PTGP	SEC	LWGP	U-SENC	LSEC
<i>PenDigits</i>	82.01 \pm 0.21	76.12 \pm 0.00	53.52 \pm 3.46	78.31 \pm 0.34	46.44 \pm 2.10	76.46 \pm 1.43	83.24 \pm 1.11	84.65 \pm 1.81
<i>USPS</i>	82.86 \pm 1.08	69.05 \pm 0.00	58.27 \pm 0.25	70.32 \pm 1.02	49.68 \pm 1.89	70.71 \pm 1.46	82.09 \pm 1.65	83.51 \pm 1.34
<i>Letters</i>	45.37 \pm 0.85	39.28 \pm 0.00	34.61 \pm 1.40	36.98 \pm 0.99	32.30 \pm 0.89	39.29 \pm 0.46	46.40 \pm 0.20	48.71 \pm 0.63
<i>MINST</i>	72.00 \pm 0.51	N/A	46.43 \pm 4.85	62.22 \pm 1.12	38.84 \pm 1.44	62.34 \pm 0.62	75.11 \pm 0.58	79.42 \pm 1.45
<i>Coverttype</i>	8.30 \pm 0.30	N/A	6.38 \pm 3.41	8.25 \pm 0.43	9.23 \pm 6.48	9.06 \pm 0.41	9.34 \pm 1.21	11.64 \pm 1.76
<i>TB-1M</i>	96.42 \pm 0.18	N/A	24.54 \pm 2.45	31.89 \pm 0.00	24.74 \pm 4.45	97.16 \pm 2.41	97.45 \pm 0.04	97.19 \pm 9.43
<i>SF-2M</i>	81.24 \pm 0.32	N/A	38.06 \pm 2.45	49.74 \pm 0.18	33.69 \pm 3.22	81.95 \pm 4.15	77.57 \pm 2.12	84.88 \pm 6.55
<i>CC-5M</i>	99.78 \pm 0.01	N/A	59.24 \pm 0.41	0.13 \pm 0.00	12.93 \pm 1.80	98.15 \pm 7.41	99.91 \pm 0.00	95.57 \pm 10.70
<i>CG-10M</i>	80.91 \pm 3.59	N/A	63.56 \pm 0.41	65.09 \pm 0.92	55.77 \pm 6.84	78.41 \pm 2.93	86.28 \pm 2.30	95.25 \pm 1.32
<i>FL-20M</i>	87.67 \pm 3.18	N/A	68.10 \pm 2.41	71.32 \pm 1.29	53.77 \pm 2.52	78.51 \pm 1.97	90.38 \pm 2.45	91.32 \pm 2.44
Avg. score	-	N/A	45.27	47.43	35.74	69.20	74.78	77.21
Avg. rank	-	6.30	5.40	4.30	5.80	3.00	1.90	1.40

Table 6: Time costs(s) of our methods and the baseline ensemble clustering methods.

Dataset	DnC-SC	EAC	KCC	PTGP	SEC	LWGP	U-SENC	LSEC
<i>PenDigits</i>	0.64	18.78	9.19	6.06	3.00	4.00	18.31	3.40
<i>USPS</i>	1.25	25.79	23.56	41.32	15.08	15.93	28.82	5.81
<i>Letters</i>	0.90	115	48.48	89.88	10.76	11.15	20.86	3.71
<i>MINST</i>	5.11	N/A	831.12	2297.05	730.33	731.64	103.35	21.36
<i>Coverttype</i>	13.15	N/A	634.18	16271.2	714.86	730.33	143.41	40.16
<i>TB-1M</i>	5.06	N/A	984.15	849.62	693.67	709.60	265.80	62.50
<i>SF-2M</i>	13.77	N/A	2225.64	1475.08	1344.66	1566.8	623.26	131.67
<i>CC-5M</i>	25.37	N/A	8541.13	3040.33	3232.06	3006	1851.2	321.71
<i>CG-10M</i>	281.05	N/A	12351.2	7244.01	7607.84	6685.8	3561.4	769.51
<i>FL-20M</i>	837.38	N/A	17112.1	13343.3	14938.73	13091	11763.07	2396.85
Avg. score	-	N/A	4276.07	4465.78	2929.10	2655.23	1837.95	375.65
Avg. rank	-	6.80	5.20	5.00	3.30	3.60	3.00	1.10

Table 7: Clustering performance (ACC(%), NMI(%), and time costs(s)) for different methods by varying number of base clusterings m .



- We set the number of landmarks as $p = 1000$ for LSEC and U-SPEC. The parameter analysis on p has been conducted in [17].
- We set the $K = 5$ for the number of nearest neighbors for LSEC and U-SPEC.
- The DnC-SC method has a unique parameter α . In the experiments, $\alpha = 50$ is used for all datasets.
- The base clusterings are generated by k -means or large-scale spectral clustering as suggested by their papers [6, 32, 9, 11, 13]. The number of cluster c of base clusterings is randomly selected from [20, 60]. The number of base clusterings is set as $m = 20$ for comparison. The parameter analysis on m will be shown in Section 5.4.
- The true number of classes on each dataset is used to conduct all experiments.
- Other parameters in the baseline methods are set as suggested by the original papers.

5.3 Comparison Results

The experimental comparison results are reported in Tables 4, 5, and 6. Note that DnC-SC is not an ensemble clustering algorithm; its clustering results are provided for reference only.

Table 8: Clustering performance (ACC(%), NMI(%), and time costs(s)) for LSEC with or without reusing of nearest landmarks.

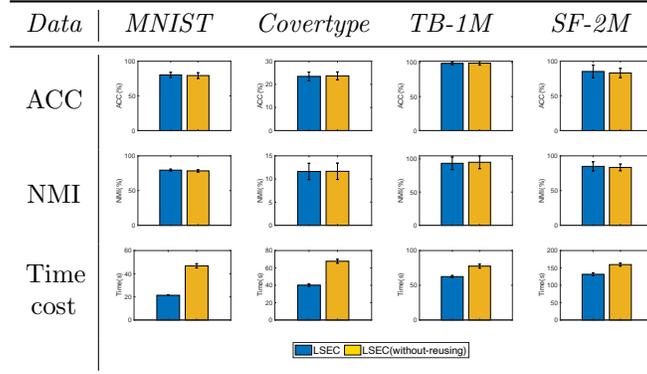
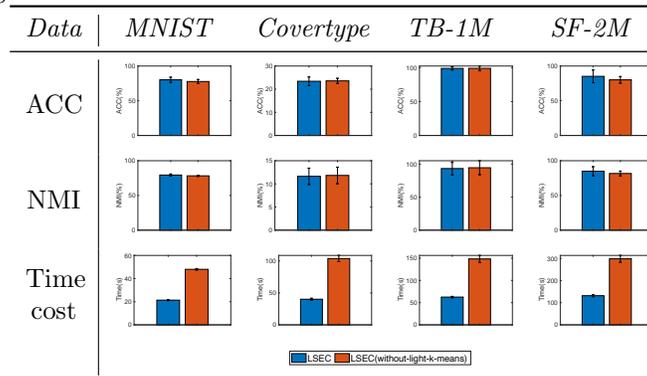


Table 9: Clustering performance (ACC(%), NMI(%), and time costs(s)) for LSEC using light- k -means or using k -means to obtain base clusterings in the ensemble generation.



As shown in Tables 4 and 5, our LSEC algorithm obtains the highest ACC and NMI scores on most of datasets. In terms of average score across the ten datasets, LSEC achieves the best average ACC(%) and NMI(%) scores of 77.01 and 77.21, respectively. While the second-best ensemble clustering method (i.e., U-SENC) achieves average ACC(%) and NMI(%) scores of 72.59 and 74.78, respectively. The EAC, KCC, PTGP, SEC, LWGP methods use the k -means based ensemble generation method. The LSEC and U-SENC methods that use the spectral clustering based ensemble generation show better clustering quality of ACC and NMI than others on most datasets. In terms of average rank, LSEC obtains an average rank of 1.90 w.r.t ACC and 1.40 w.r.t. NMI, while the second-best method obtains an average rank of 2.30 w.r.t. ACC and 1.90 w.r.t. NMI.

In Table 6, the time costs of different ensemble clustering methods are provided. The proposed LSEC method achieves the lowest time costs on nine datasets and the second-lowest time cost on one dataset. Except *PenDigits* dataset, LSEC is 2.4 (*FL-20M*) to 5.75 (*CC-5M*) times ahead of the second-best method in time consumption. The LSEC method has shown its significant advantage over other ensemble clustering methods, especially on large-scale datasets.

5.4 Parameter analysis on Ensemble Size m

We conduct a parameter analysis experiment to demonstrate the performance of the proposed method, varying different parameter values of m . The parameter m denotes the number of base clusterings, which is a common parameter in all ensemble clustering methods. We select four dataset (*MNIST*, *Covertypes*, *TB-1M* and *SF-2M*) as benchmark datasets to conduct the following experiments. As shown in Table 7, LSEC shows better performance of ACC and NMI than most other ensemble clustering methods except ACC score on *Covertypes* dataset. Meanwhile, LSEC consistently requires a lower computational cost than all other ensemble clustering methods.

5.5 Influence of reusing of K -nearest landmarks

In this section, we compare the performances of the proposed method with or without reusing of nearest landmarks, denoted as LSEC and LSEC-without-reusing. The experimental results are reported in Table 8. As we mentioned, the reusing of nearest landmarks brings better efficiency in searching K -nearest landmarks. In Table 8, LSEC and LSEC-without-reusing show similar performances to each other, but LSEC cost obviously less time. Since reusing of nearest landmarks does not influence the accuracy of nearest landmarks, we consider that the difference of ACC and NMI between the two methods comes from the randomness of the algorithm. This result indicates that reusing of nearest landmarks achieves significantly better efficiency while maintaining a similar clustering result.

5.6 Influence of light- k -means

In this section, we compare the performances of the proposed method using light- k -means or using k -means to obtain base clusterings in the ensemble generation. The experimental results are reported in Table 9. Generally, two methods show the similar performance of ACC and NMI. Especially, LSEC achieves slightly better ACC and NMI on *MNIST* and *SF-2F* datasets, which is possible because the light- k -means method can provide better diversity of base clusterings on these datasets. Overall, using light- k -means in ensemble generation significantly improves the efficiency of LSEC and yields similar clustering quality compared to the k -means method.

6 Conclusion

In this paper, we propose a large-scale spectral ensemble clustering (LSEC) method to strike a better balance between the efficiency and effectiveness of ensemble clustering on large-scale datasets. We design an efficient ensemble generation framework to produce based clustering, applying divide-and-conquer large-scale spectral clustering to find high-quality base clusterings. In the ensemble generation of the proposed method, we accelerate the process of searching K -nearest neighbors by reusing strategy and obtaining base clustering by the light- k -means method. After the ensemble generation step, we combine all based clustering into a consensus cluster through a bipartite graph partitioning based consensus function. The proposed method achieves lower computational complexity than most existing ensemble clustering methods. Experiments conducted on ten large-scale datasets show that the proposed method outperforms other state-of-the-art large-scale spectral clustering methods.

Acknowledgment

This study was supported by the New Energy and Industrial Technology Development Organization (NEDO) Grant (ID:18065620) and JST COI-NEXT.

References

- [1] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.
- [2] Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.
- [3] Deng Cai, Xiaofei He, and Jiawei Han. Speed up kernel discriminant analysis. *The VLDB Journal*, 20(1):21–33, 2011.

- [4] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.
- [5] Xiaoli Zhang Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, page 36, 2004.
- [6] Ana LN Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):835–850, 2005.
- [7] Peter W Frey and David J Slate. Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6(2):161–182, 1991.
- [8] Dong Huang, Jian-Huang Lai, and Chang-Dong Wang. Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing*, 170:240–250, 2015.
- [9] Dong Huang, Jian-Huang Lai, and Chang-Dong Wang. Robust ensemble clustering using probability trajectories. *IEEE transactions on knowledge and data engineering*, 28(5):1312–1326, 2015.
- [10] Dong Huang, Jianhuang Lai, and Chang-Dong Wang. Ensemble clustering using factor graph. *Pattern Recognition*, 50:131–142, 2016.
- [11] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. Locally weighted ensemble clustering. *IEEE transactions on cybernetics*, 48(5):1460–1473, 2017.
- [12] Dong Huang, Chang-Dong Wang, Hongxing Peng, Jianhuang Lai, and Chee-Keong Kwoh. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [13] Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226, 2019.
- [14] Natthakan Iam-On, Tossapon Boongeon, Simon Garrett, and Chris Price. A link-based cluster ensemble approach for categorical data clustering. *IEEE Transactions on knowledge and data engineering*, 24(3):413–425, 2010.
- [15] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483, 2017.

- [16] Hongmin Li, Xiucai Ye, Akira Imakura, and Tetsuya Sakurai. Ensemble learning for spectral clustering. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1094–1099. IEEE, 2020.
- [17] Hongmin Li, Xiucai Ye, Akira Imakura, and Tetsuya Sakurai. Divide-and-conquer based large-scale spectral clustering. *arXiv preprint*, page 2104.15042, 2021.
- [18] Tao Li, Mitsunori Ogihara, and Sheng Ma. On combining multiple clusterings. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 294–303, 2004.
- [19] Hongfu Liu, Tongliang Liu, Junjie Wu, Dacheng Tao, and Yun Fu. Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 715–724, 2015.
- [20] Hongfu Liu, Junjie Wu, Tongliang Liu, Dacheng Tao, and Yun Fu. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE transactions on knowledge and data engineering*, 29(5):1129–1143, 2017.
- [21] Hongfu Liu, Rui Zhao, Hongsheng Fang, Feixiong Cheng, Yun Fu, and Yang-Yu Liu. Entropy-based consensus clustering for patient stratification. *Bioinformatics*, 33(17):2691–2698, 2017.
- [22] Murilo Coelho Naldi, ACPLF Carvalho, and Ricardo JGB Campello. Cluster ensemble selection based on relative validity indexes. *Data Mining and Knowledge Discovery*, 27(2):259–289, 2013.
- [23] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in neural information processing systems*, pages 617–623, 2000.
- [24] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.
- [25] Aditya Tandon, Aiiad Albeshri, Vijey Thayanathan, Wadee Alhalabi, and Santo Fortunato. Fast consensus clustering in complex networks. *Physical Review E*, 99(4):042301, 2019.
- [26] Zhiqiang Tao, Hongfu Liu, Sheng Li, and Yun Fu. Robust spectral ensemble clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 367–376. ACM, 2016.
- [27] Alexander Topchy, Anil K Jain, and William Punch. Combining multiple weak clusterings. In *Third IEEE international conference on data mining*, pages 331–338. IEEE, 2003.

- [28] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.
- [29] Chen Wang, Sebastian M Armasu, Kimberly R Kalli, Matthew J Maurer, Ethan P Heinzen, Gary L Keeney, William A Cliby, Ann L Oberg, Scott H Kaufmann, and Ellen L Goode. Pooled clustering of high-grade serous ovarian cancer gene expression leads to novel consensus subtypes associated with survival and surgical outcomes. *Clinical Cancer Research*, 23(15):4077–4085, 2017.
- [30] Fei Wang, Xin Wang, and Tao Li. Generalized cluster aggregation. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.
- [31] Xi Wang, Chunyu Yang, and Jie Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009.
- [32] Junjie Wu, Hongfu Liu, Hui Xiong, Jie Cao, and Jian Chen. K-means-based consensus clustering: A unified view. *IEEE transactions on knowledge and data engineering*, 27(1):155–169, 2014.
- [33] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.
- [34] Donghui Yan, Ling Huang, and Michael I Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907–916. ACM, 2009.
- [35] Xiucui Ye and Tetsuya Sakurai. Robust similarity measure for spectral clustering based on shared neighbors. *ETRI journal*, 38(3):540–550, 2016.
- [36] Xiucui Ye and Tetsuya Sakurai. Spectral clustering with adaptive similarity measure in kernel space. *Intelligent Data Analysis*, 22(4):751–765, 2018.
- [37] Li Zheng, Tao Li, and Chris Ding. A framework for hierarchical ensemble clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(2):1–23, 2014.
- [38] Caiming Zhong, Xiaodong Yue, Zehua Zhang, and Jingsheng Lei. A clustering ensemble: Two-level-refined co-association matrix with path-based transformation. *Pattern Recognition*, 48(8):2699–2709, 2015.