Divide-and-conquer based Large-Scale Spectral Clustering

Hongmin Li*, Xiucai Ye, Akira Imakura and Tetsuya Sakurai

Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan

ARTICLE INFO

Keywords:
Spectral Clustering
Landmark selection
Approximate Similarity Computation
Large-scale clustering
Large-scale datasets

ABSTRACT

Spectral clustering is one of the most popular clustering methods. However, how to balance the efficiency and effectiveness of the large-scale spectral clustering with limited computing resources has not been properly solved for a long time. In this paper, we propose a divide-and-conquer based large-scale spectral clustering method to strike a good balance between efficiency and effectiveness. In the proposed method, a divide-and-conquer based landmark selection algorithm and a novel approximate similarity matrix approach are designed to construct a sparse similarity matrix within low computational complexities. Then clustering results can be computed quickly through a bipartite graph partition process. The proposed method achieves the lower computational complexity than most existing large-scale spectral clustering methods. Experimental results on ten large-scale datasets have demonstrated the efficiency and effectiveness of the proposed method. The MATLAB code of the proposed method and experimental datasets are available at https://github.com/Li-Hongmin/MyPaperWithCode.

1. Introduction

Clustering is one of the most fundamental problems in data mining and machine learning, aiming to categorize data points into clusters such that the data points in the same cluster are more similar while data points in different clusters are more different from each other [20, 24, 16]. Spectral clustering has attracted increasing attention due to the promising ability to deal with nonlinearly separable datasets [8, 17]. It has been successfully applied to various problem domains such as biology [18], image segmentation [31], and recommend systems [33, 28]. Although spectral clustering algorithm often provides better performances than traditional clustering algorithm like K-means especially for complex datasets, it is significantly limited to be applied to large-scale datasets due to its high computational complexity and space complexity [13, 27].

The conventional spectral clustering algorithm mainly consists of two high-cost steps, i.e., similarity matrix construction and eigen-decomposition. For a dataset with \( N \) objects, the two steps take computational complexities of \( O(N^2d) \) and \( O(N^3) \), respectively. The computational consumption of these two steps is the main reason that hinders the application of spectral clustering algorithms on large-scale data.

In recent years, there has been an increasing amount of literature on alleviating the computational complexity of spectral clustering [9, 4, 14, 3, 32, 27, 11, 12]. Previous research [7] has established that the sparse similarity matrix constructed by only remaining \( k \)-nearest neighbors or \( e \)-nearest neighbors can efficiently reduce the space complexity. As a result, some sparse eigensolvers can solve the eigen-decomposition problems within the lower computational complexity. The matrix specification strategy can avoid storing the dense similarity matrix to reduces the space complexity, but it still needs to compute the dense similarity matrix at first, which costs \( O(N^2d) \) computational complexity. Besides the matrix specification, another commonly used strategy is based on a cross-similarity matrix construction [9, 4, 14, 27, 12, 11]. Fowlkes et al. [9] apply the Nyström method to reduce the high complexity of spectral clustering algorithm, which first randomly selects a small subset of samples as landmarks, then construct a similarity sub-matrix between these landmarks and remaining samples. Although the random landmark selection is very efficient, it is often unstable concerning the quality of the landmark set. Moreover, it has been shown that a larger \( p \) is often favorable for better approximation. To address the potential instability of random selection, Cai et al. [4] extend the Nyström method and propose a landmark based large-scale spectral clustering (LSC) method, which uses \( k \)-means to obtain \( p \) cluster centers as \( p \) landmark points to construct the similarity sub-matrix. With the constructed \( N \times p \) sub-matrix, they then convert it into sparse by preserving the \( k \)-nearest landmarks of the data points and filling with zeros to others. By the \( k \)-means based landmarks selection, the LSC algorithm shows better performance than Nyström. On this basis, some studies [30, 3, 32, 19, 27, 12] on the landmark selection are further proposed to improve the instability of sub-matrix based large-scale spectral clustering. However, the computational complexity of the sub-matrix construction can still be a critical bottleneck when dealing with large-scale clustering tasks. Huang et al. [11] propose a hybrid representative (landmark) selection method that initializes candidate samples randomly from the dataset and performs \( k \)-means to obtain \( p \) cluster centers as the representatives then computes the approximation of \( K \)-nearest representatives. It does not compute the dense similarity sub-matrix but approximates a sparse sub-matrix, further reducing similarity construction costs. However, those sub-matrix based spectral clustering algorithms are typically restricted by an \( O(Np) \) or \( O(Np^{\frac{3}{2}}) \) complexity bottleneck, which is still a
critical hurdle for them to deal with large-scale datasets where a larger \( p \) is often desired for achieving better approximation. Although some considerable studies have been proposed in recent years, it remains a highly challenging problem, i.e., how to make spectral clustering handle large-scale datasets efficiently and effectively within limited computing resources.

In this paper, to achieve a better balance between the effectiveness and efficiency of the spectral clustering for large-scale datasets, we propose the divide-and-conquer spectral clustering (DnC-SC) method. In DnC-SC, a novel divide-and-conquer based landmark selection method is proposed to generate high-quality \( p \) landmarks, which reduces the computational complexity of \( k \)-means based selection from \( O(Npdt) \) to \( O(Npd) \), where \( \alpha \) is the selection rate parameter that determines the upper bound of computational complexity. Besides, a fast approximation method for \( K \)-nearest landmarks is designed to efficiently build a sparse sub-matrix with \( O(NKd) \) computational complexity and \( O(NKd) \) space complexity. A cross similarity matrix is constructed between the data points and the \( p \) landmarks, which can be interpreted as the edges matrix of a bipartite graph. The bipartite graph partitioning is then conducted to solve the spectrum with \( O(NK(K + k) + p^3) \), where \( k \) is the number of clusters. Finally, the \( k \)-means method is used to obtain the clustering result on the spectrum with \( O(Nk^2t) \), where \( t \) is the number of iterations during \( k \)-means. As it generally holds that \( k, K, \alpha \ll p \ll N \), the computational and space complexity of our DnC-SC algorithm are respectively dominated by \( O(Npd) \) and \( O(NK) \). The experimental results on ten large-scale datasets (consisting of five real-word datasets and five synthetic datasets) show the priority performance of proposed methods on both efficiency and effectiveness.

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

- A divide-and-conquer-based landmark selection method is proposed to efficiently find \( p \) centralized subset centers as landmarks in a recursive manner.

- A fast \( K \)-nearest landmarks search method is designed, which uses centers’ nature of landmarks to identify the most possible \( K \)-nearest landmarks candidates.

- A large-scale spectral clustering algorithm termed DnC-SC is proposed, which efficiently constructs the similarity matrix and uses bipartite graph partitioning to obtain final clustering results. Its computational and space complexity is dominated by \( O(Npd) \) and \( O(NK) \), which achieves a lower computational complexity than most existing large-scale spectral clustering methods.

### 2. Preliminaries

This section reviews the literature related to spectral clustering and large-scale spectral clustering extensions.

#### 2.1. Spectral Clustering

Spectral clustering aims to partition the data points into \( k \) clusters using the spectrum of the graph Laplacians \([23]\). Given a dataset \( X = \{x_1, \ldots, x_N\} \) with \( N \) data points, spectral clustering algorithm first constructs similarity matrix \( W \), where \( w_{ij} \) indicates the similarity between data points \( x_i \) and \( x_j \) via a similarity measure metric.

Let \( L = D - W \), where \( L \) is called graph Laplacian and \( D \) is a diagonal matrix with \( d_{ij} = \sum_{j=1}^{n} w_{ij} \). The objective function of spectral clustering can be formulated based on the graph Laplacian as follow:

\[
\min_{U} \text{tr} \left( U^T LU \right), \quad \text{s.t.} \quad U^T U = I,
\]

where \( \text{tr}(\cdot) \) denotes the trace norm of a matrix. The rows of matrix \( U \) are the low dimensional embedding of the original data points. Generally, spectral clustering computes \( U \) as the bottom \( k \) eigenvectors of \( L \), and finally applies \( k \)-means on \( U \) to obtain the clustering results.

#### 2.2. Large-scale Spectral Clustering

##### 2.2.1. Similarity Sub-matrix construction

Instead of an \( N \times N \) similarity matrix, many large-scale spectral clustering methods \([9, 4, 14, 27, 12, 11]\) are using a similarity sub-matrix to represent each data points. The similarity sub-matrix consists of the cross-similarities between data points and a set of representative data points (i.e., landmarks) via some similarity measures, as

\[
B = \Phi(X, R),
\]

where \( R = \{r_1, r_2, \ldots, r_p\} \) \((p \ll N)\) is a set of landmarks with the same dimension to \( X \), \( \Phi(\cdot) \) indicate a similarity measure metric, and \( B \in \mathbb{R}^{N \times p} \) is the similarity sub-matrix to represent the \( X \in \mathbb{R}^{N \times d} \) with respect to the \( R \in \mathbb{R}^{p \times d} \).

Ideally, the landmark points \( r_1, r_2, \cdots, r_p \) would roughly represent the distribution of \( X \). Some previous studies \([4]\) show the effectiveness of \( k \)-means based selection. The objective function of \( k \)-means based landmark selection can represented as follows:

\[
R = \arg \min_{r_1, \cdots, r_p} \sum_{i=1}^{p} \sum_{j \in S_i} \left\| x_j - r_i \right\|^2,
\]

where \( S_1, S_2, \ldots, S_p \) indicate the subsets that are nearest to \( r_1, r_2, \cdots, r_p \), respectively. However, directly conducting \( k \)-means on large-scale datasets faces a high time cost of \( O(Npdt) \). Moreover, \( k \)-means often needs more iterations to converges on large-scale datasets.

##### 2.2.2. Efficient Bipartite Graph Partitioning

The similarity sub-matrix \( B \) reflects the relationship between \( X \) and \( R \), which can be naturally treated as a bipartite graph \( G = \{X, R, B\} \). The goal of bipartite graph partitioning is to partition the graph \( G \) into \( k \) groups. The full similarity matrix of \( G \) is then designed as \([29]\)

\[
W = \begin{bmatrix}
0 & B \\
B^T & 0
\end{bmatrix}.
\]
Divide-and-conquer based Large-Scale Spectral Clustering

The size of matrix $W$ is $(N + p) \times (N + p)$. The conventional spectral clustering finds a low dimensional embedding via the spectrum of graph Laplacian, which solves the generalized eigen-problem [21]:

$$Lf = \gamma Df,$$

where $L = D - W$ is the graph Laplacian and $D$ is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}$. Note that the eigenvector $f$ can be interpreted as two parts $u \in \mathbb{R}^{N}$ and $v \in \mathbb{R}^{p}$.

$$f = \begin{bmatrix} u \\ v \end{bmatrix},$$

where $u$ is the eigenvector on $X$ side while $v$ is the eigenvector on $R$ side.

The problem is how to efficiently compute eigenvector $u$ and construct a low dimensional embedding on original data $X$. An efficient computation method called transfer cut is often used to compute the spectrum of graph Laplacian for bipartite graph partitioning problems. Instead of directly computing $f$ by partial SVDs or dual property of SVD [4], the transfer cuts process first computes the $v$ by solving a much smaller eigen-problem as follows:

$$L_{RV} = \lambda D_{RV},$$

where $L = D - B^{T}D_{X}^{-1}B$, $D_{X} \in \mathbb{R}^{n \times n}$ 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. It has been pointed out that the eigen-problems (5) on original bipartite graph $G$ and the much smaller one (7) are essential equivalence [15]. Let $\{(\lambda_{1}, v_{1}), \ldots, (\lambda_{k}, v_{k})\}$ be the bottom $k$ eigenvairs of the eigen-problem (7) and $0 = \lambda_{1} \leq \cdots \leq \lambda_{k} < 1$. Then $\{(\gamma_{1}, f_{1}), \ldots, (\gamma_{k}, f_{k})\}$ are the bottom $k$ eigenvairs of the eigen-problem (5) and $0 = \gamma_{1} \leq \cdots \leq \gamma_{k} < 1$. It have been proved that [14]

$$u_{i} = \frac{1}{1 - \gamma_{i}} T v_{i},$$

where $1 \leq \gamma_{i} < 1$, $\lambda_{i} = \gamma_{i}(2 - \gamma_{i})$ and $T := D_{X}^{-1}B$ is called the associated transition probability matrix. Therefore bottom $k$ eigenvectors $u_{1}, \ldots, u_{k}$ are calculated according to (7) and (8). Let $U \in \mathbb{R}^{N \times k}$ be the matrix containing the vectors $u_{1}, \ldots, u_{k}$ as columns. Then $U$ is the spectral embedding of the large-scale spectral clustering algorithm. Finally, $k$-means is conducted on the embedding to obtain final clustering results.

3. Proposed Framework

To further reduce the complexity of spectral clustering, we propose the DnC-SC method that complies with the sub-matrix based formulation [9, 4] and aims to break through the efficiency bottleneck of previous algorithms. DnC-SC consists of three phases: (1) Divide-and-conquer based landmark selection: we consider landmark selection as an optimization problem and present a divide-and-conquer based landmark selection method to find the landmarks via solving the sub-optimization problems recursively. (2) Approximate similarity matrix construction: we design a novel strategy to efficiently approximate the $K$-nearest landmarks for each data point and construct a sparse cross-similarity matrix between the $N$ data points and the $p$ landmarks. (3) Bipartite graph partitioning: we interpret the cross-similarity as a bipartite graph and conduct the bipartite graph partitioning to obtain the clustering result. We summarize the proposed method in Figure 1.

3.1. Divide-and-conquer based Landmark Selection

We propose a divide-and-conquer based landmark selection method, which aims to find a set of high-quality landmarks efficiently. Instead of directly dividing data points into $p$ subsets like (3) for landmark selection, we first divide data points into $a$ subsets, then recursively divide each subset into $k$ smaller subsets ($k \leq a$), until the total number of subsets reaches $p$. Denote $\alpha$ as a small number and $a \ll p \ll n$. We define $a$ as selection rate parameter that is the upper boundary of desired subset number in each dividing process to limit the computational complexity. Figure 2 gives a simple example. The data points are recursively divided into $a$ subsets until the total number of subsets is $p$, which avoids directly applying $k$-means to obtain too many subsets at once.

In the divide-and-conquer strategy, the number of desired subsets in each iteration is much smaller than $p$, and the subsets are smaller and smaller during iterations than directly applying $k$-means to datasets. Suppose each dividing process will divide $a$ subsets of the same size, the number of subsets increases exponentially and reaches $p$ subsets in $\lceil \log_{a} p \rceil$ rounds. Denote $S_{t}^{(1)} = X$ is the initial subset. In first round of dividing process, $N$ data points are divided into $a$ subsets $S_{t}^{(2)}$, $S_{t}^{(3)}$, ..., $S_{t}^{(2)}$ with computational complexity of $O(N a dt)$, which is illustrated in the (1) of Figure 2. There are $a$ subsets currently, each current subset is then divided into new subsets in the second round with computational complexity of $O(N a dt)$. For example, $S_{t}^{(2)}$ is further divided into new subsets $S_{t}^{(3)}$, $S_{t}^{(3)}$, ..., $S_{t}^{(3)}$ with computational complexity of $O(N a dt)$.

The dividing process of second round generates $a^{2}$ new subsets $S_{t}^{(3)}$, $S_{t}^{(3)}$, ..., $S_{t}^{(3)}$ totally. Similarly, the computational complexity of any $i$-th ($i = 1, 2, \ldots, \lceil \log_{a} p \rceil$) dividing round will be $O(a^{i-1} \frac{N}{a^{i}} dt) = O(N dt)$. Therefore, the total computational complexity of divide-and-conquer strategy is $O(\lceil \log_{a} p \rceil N dt)$. Note that $\log_{a} p$ is a small value and can be treated as a constant, e.g., $\log_{a} p = 1.77$ when setting $p = 1000$ and $a = 50$. We further simplify the total computational cost as $O(N dt)$. Compared with $k$-means based landmark selection, the divide-and-conquer strategy can naturally reduce the computational complexity from $O(N p dt)$ to $O(N dt)$ ($a \ll p$). Moreover, we design an efficient dividing algorithm, named light-$k$-means, to further accelerate...
Divide-and-conquer based Large-Scale Spectral Clustering

Figure 1: An overview of proposed DnC-SC method. Given a dataset, the DnC-SC method first finds the landmarks via divide-and-conquer based landmark selection, then approximately constructs the similarity matrix, finally conducts the bipartite graph partitioning to obtain final clustering results. Our main contributions focus on the first two phases (colored as orange), i.e., the landmark selection and similarity construction phases.

Figure 2: An illustration of the divide-and-conquer base landmark selection: (1) The dataset is initially divided into $k_1^{(1)} = 3$ initial subsets; (2) Each subset is further divided into 3 smaller subsets and the total number of subsets reaches $p = 9$. Finally, the centers of subsets are turned as landmarks.

3.1.1. Divide-and-conquer Selection Strategy

Before starting landmark selection, we first review the optimization problem (3). The variables in (3) are the $r_1, \ldots, r_p$ which are used to map the unique $S_1, \ldots, S_p$. By setting the subsets $S_1, \ldots, S_p$ and landmark number as the variables, we can rewrite the optimization problem (3) into a function form as follows:

$$g(X, p) = \arg \min_{S_1, \ldots, S_p} \sum_{i=1}^{p} \sum_{s_i \in S_i} \| x_i - r_s \|^2,$$  

where $g()$ indicates a centralized clustering problem that divides $X$ into $p$ subsets and $r_s$ is the center of subset $S_i$. For any dividing problem that divides $Q$ into $h$ subsets, the function $g(Q, h)$ can be used to describe the dividing problem, and its computational complexity is $O(||Q||hdt)$, where $||Q||$ is the total number of samples in $Q$. More importantly, function $g(Q, h)$ can be used to derive the recursive function as follows:

$$g(Q, h) = \bigcup_{i=1}^{m} g(A_i, k_i), \quad (10)$$

$$\{A_1, \ldots, A_m\} = g(Q, m), \quad (11)$$

where $A_i$ is a subset of $Q$ and $Q = \bigcup_{i=1}^{m} A_i$; $k_i$ is the desired subset number of subset $A_i$ and $h = \sum_{i=1}^{m} k_i; m < h$. (10) can simply divide any optimization problem into $m$ sub-problems, which builds a bridge between global problem $g(Q, h)$ and local problem $g(A_i, k_i)$. We can recursively apply (10) and (11) to divide the optimization problem (9) into the sub-problems small enough and solve them locally and efficiently.

Denote $c_j$ as the total number of subsets during $i$-th iteration. We will stop the recursive process in the $j$-th iteration when $c_j$ reaches the desired total number of subsets $p$. Initially, we assign all data points as one subset. As we only have one subset ($c_1 = 1 < p$), the dividing process happens. Let $k_1^{(i)}$ be the desired number of subsets for dividing process on $j$-th subset during $i$-th iteration. We naturally set the desired number $k_1^{(i)} = p$. However, directly apply $g(X, p)$ may be time-consuming. For $k > a$, we force $k = a$ to obtain subsets partially. As a result, we have the initial setting as follows:

$$\begin{cases} S_1^{(1)} = X \\ k_1^{(1)} = a \end{cases} \quad (12)$$

In the first iteration, we divide $S_1^{(1)}$ into $k_1^{(1)}$ subsets in as follows:

$$\{S_1^{(2)}, S_2^{(2)}, \ldots, S_c^{(2)}\} = g(S_1^{(1)}, k_1^{(1)}), \quad (13)$$

where $S_1^{(1)}$ indicates the $i$-th subset during first iteration and $c_2 = \sum_{i=1}^{c_1} k_1^{(i)}$ is the total number of subsets.

From the second iteration, there are more and more subsets being obtained. Thus, we need a subset number allocation strategy to determine the desired number of subsets and guide iteration dynamically. We define the residual sum of
squares (RSS) of subset \(S_i^{(j)}\) as
\[
    c_i^{(j)} = \sum_{x_j \in S_i^{(j)}} \|x_j - r_i\|^2,
\]
where \(r_i\) is the center of the subset \(S_i^{(j)}\). Consider the global problem (3), the desired number of subsets of the RSS. We propose a dynamical allocation strategy as follows:

\[
k_i^{(j)} = \begin{cases} 
    \frac{c_i^{(j)}}{\sum_{i} c_i^{(j)}} \cdot p, & \text{if } \frac{\sum_{i} c_i^{(j)}}{\sum_{i} p} < \alpha, \\
    \alpha, & \text{otherwise},
\end{cases}
\]

where \(k_i^{(j)}\) is the allocated dividing number for subset \(S_i^{(j)}\). Then, all \(k_i^{(j)}\) are turned as integers and fix the \(c_{j+1} = \alpha p\), where \(c_{j+1} = \sum_i k_i^{(j)}\). After obtaining \(k_1^{(j)}, ..., k_c^{(j)} (c < p)\), we will divide each subset \(S_i^{(j)}\) into \(k_i^{(j)}\) smaller subsets via \(g(S_i^{(j)}, k_i^{(j)})\). We then collect all subsets as follows:

\[
    \left( S_1^{(j+1)}, S_2^{(j+1)}, ..., S_{c_j+1}^{(j+1)} \right) = \bigcup_{i=1}^{c_j} g(S_i^{(j)}, k_i^{(j)}).
\]

We repeat the above process until \(p\) subsets have been produced and set the \(p\) subset centers as the landmarks.

Take an example using Figure 2, where we set \(\alpha = 3\) and \(p = 9\). In the first iteration, we assign all data points as one subset. Since the desired landmark number \(p > \alpha\), we set \(k_1^{(1)} = 3\). Then we initially divide the dataset \(S_1^{(1)} = X\) into \(k_1^{(1)} = 3\) subsets. In the second iteration, there are three subsets \(S_2^{(1)}, S_2^{(2)}\) and \(S_2^{(3)}\). According to (15), we compute the \(k_2^{(1)}, k_2^{(2)}, k_2^{(3)}\). Suppose \(k_1^{(2)} = k_2^{(2)} = k_2^{(3)} = 3\), we then divide \(S_3^{(2)}, S_3^{(2)}, S_4^{(2)}\) into \(k_2^{(1)}, k_2^{(2)}\) and \(k_2^{(3)}\) smaller subset respectively. In the third iteration, there are 9 subsets \(S_5^{(1)}, S_5^{(2)}, S_5^{(3)}\). Since the total number of subsets \(c_3 = 9\) reaches the desired landmark number \(p = 9\), we stop the recursive process in the third iteration. Finally, compute the subset centers \(r_1, r_2, ..., r_9\) of \(S_1^{(3)}, S_2^{(3)}, S_3^{(3)}\) and set them as the landmarks.

Note that the dividing process \(g(\cdot)\) can be directly solved by the \(k\)-means method. However, directly apply \(k\)-means on large data is time-consuming. To further reduce the complexity, we propose a modified \(k\)-means method, named light-

\[ k\]-means. When dataset size is large, we conduct the dividing process via light-

\[ k\]-means. Otherwise, we use the traditional \(k\)-means method. We summary the divide-and-conquer based landmark selection method in Algorithm 1.

### 3.1.2. Light-k-means Algorithm

We define \(p'\) as an upper bound. When the size of \(S_i^{(j)}\) is larger than \(p'\), we will use light-

\[ k\]-means to compute \(g(S_i^{(j)}, k_i^{(j)})\). The light-

\[ k\]-means is performed as the following steps:

1. Randomly select \(p'\) representatives from \(S_i^{(j)}\) and denote them in a set as \(H\) and the complement of \(H\) is \(H^c\).

\[
    \textbf{Algorithm 1: Divide-and-conquer based landmark selection method}
\]

**Input:** Dataset \(X\), the number of landmarks \(p\), selection rate \(\alpha\);

**Output:** Landmarks \(R\);

1. Initialize: Set the \(S_1^{(1)}\) and \(k_1^{(1)}\) via (12);
2. \(c_1 = 1\);
3. \(j = 1\);
4. while total number of subsets \(c_j < p\) do
   5. Set the \(k_1^{(j)}, ..., k_j^{(j)}\) via (15);
   6. foreach \(S_i^{(j)}\) do
      7. if the size of \(S_i^{(j)}\) is larger than \(p'\) then
         8. Conduct (16) via light-
      else
         10. Conduct (16) via \(k\)-means;
     end
   12. \(c_{j+1} = \sum_i k_i^{(j)}\);
   13. \(j = j + 1\);
16. Collect the latest cluster centers as \(R\).

2. Conduct \(k\)-means to divide \(H\) into \(k_i^{(j)}\) subsets;
3. Find the nearest subset centers for the remained data points in \(H^c\);
4. Assign the remained data points in \(H^c\) to their nearest subsets (with the center nearest to these points).

Figure 3 shows a comparison between \(k\)-means and light-

\[ k\]-means method, which are the implementation examples of (1) in Figure 2. Given a subset \(S_1^{(1)}\), the \(k\)-means first randomly select \(p'\) data points and denotes them as \(H\) and the complement is \(H^c\) \((S_1^{(1)} = H \cup H^c)\). Then the \(k\)-means is used to divide \(H\) into \(k_i^{(j)}\) subsets, i.e., \(A_1, A_2, A_3\). For each data points in \(H^c\), find its nearest center and assign it to the subset, i.e., \(A_1^c, A_2^c, A_3^c\) according to its nearest center. Finally, return the combined subsets \(S_1^1, S_2^2, S_3^3\) as the results of this dividing process.

\[
    \textbf{Algorithm 2: Light-k-means}
\]

**Input:** Data \(S_i^{(j)}\), the number of cluster \(k_i^{(j)}\), number of samples \(p'\);

**Output:** \(k_i^{(j)}\) subsets;

1. Randomly select \(p'\) samples from \(S_i^{(j)}\) and denote them as \(H\);
2. Denote the complement of \(H\) as \(H^c\);
3. Apply \(k\)-means to divide \(H\) into \(k_i^{(j)}\) subsets;
4. Find the nearest center of samples in \(H^c\);
5. Assign the samples in \(H^c\) to the subset according to their nearest centers.

Denote \(n_i^{(j)}\) as the number of samples in \(S_i^{(j)}\). The computational complexity of light-

\[ k\]-means for the dividing pro-
Divide-and-conquer based Large-Scale Spectral Clustering

![Diagram](image)

**Figure 3:** An comparison between k-means and light-k-means. (a) k-means directly divide all samples $S^{(j)}_i$ into 3 subsets. (b) In light-k-means, k-means is applied on $p'$ representatives, which significantly reduces the complexity on large data.

![Diagram](image)

**Figure 4:** Comparison of the landmarks produced by (a) random selection, (b) k-means based selection, and (c) Divide-and-conquer based selection.

process $g(S^{(j)}_i, k^{(j)}_i)$ should be $O(p'k^{(j)}_i)$, $O((n^{(j)}_i - p')k^{(j)}_i)$, and $O(n^{(j)}_i k^{(j)}_i d + p'k^{(j)}_i d (t - 1))$, where $O(n^{(j)}_i k^{(j)}_i d)$ is the dominant term. While, k-means costs $O(n^{(j)}_i k^{(j)}_i d)$ for the same dividing process. Compared with k-means, light-k-means significantly alleviates the computational complexity of iterative optimization. Empirically, the number of $p'$ is suggested to be several times larger than $p$, e.g., $p' = 10p$, to provide enough samples for the k-means algorithm. Since our landmark selection focuses more on the local dividing, the light-k-means can effectively divide the large subsets into small ones and find more accurately subsets locally. Finally, we summarise the light-k-means method in Algorithm 2.

By introducing the divide-and-conquer based landmark selection, the complexity of landmark selection is reduced to $O(Nad)$ from $O(Npdt)$ of k-means based selection. Figure 4 illustrates that the proposed divide-and-conquer based landmark selection can better represent data distribution than the random selection and has similar performance k-means based selection, but it has the lower complexity than k-means based selection.

**3.2. Approximate Similarity Matrix Construction**

After landmark selection, the next object is to construct a similarity matrix between data points and the landmarks. Instead of dense similarity matrix, we design a similarity matrix $B \in \mathbb{R}^{N \times p}$ according to K-nearest neighbor as follows:

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

(17)

where $N_K(x_i)$ denotes the set of K-nearest landmarks of $x_i$ and $\sigma$ is the bandwidth of Gaussian kernel. Note that there are only $NK$ non-zero entries in the sparse matrix $B$.

To estimate $N_K(x_i)$, we propose a new K-nearest landmarks approximation method. The main idea is to use the subset centers’ nature of landmarks to estimate the possible nearest candidates, as shown in Figure 5. Formally, we denote $S_{x_i}$ as the subset that $x_i$ belongs to, and the landmark $r_{x_i}$ as the center of $S_{x_i}$. Since $r_{x_i}$ is the subset center of $x_i$, it essentially is the nearest landmark of $S_{x_i}$ according to (3). Take the advantage of this landmark nature, we search the K-nearest landmarks of each data point $x_i \in X$ according to the following two steps:

**Step 1: Find $K'$ possible candidates.** As (1) of Figure 5 shows, we find the K'-nearest ($K' > K$) landmarks of $r_{x_i}$ and denoted them as $N_K'(r_{x_i}) = \{r_{x_i}', \ldots, r_{K'}^{x_i}\}$. Since the exact K-nearest landmarks of $x_i$ are highly possible closed to $r_{x_i}$, we treat $N_K'(r_{x_i})$ as possible candidates set. Empirically, the number of $K'$ is suggested to be several times larger than $K$, e.g., $K' = 10K$, to provide enough candidates to search $N_K(x_i)$.

![Diagram](image)

**Figure 5:** An illustration of our K-nearest landmarks approximation: (1) Find the K'-nearest ($K' > K$) landmarks of $r_{x_i}$, where $x_i \in S_{x_i}$ and $r_{x_i}$ is the subset center of $S_{x_i}$; (2) Find the K-nearest landmarks among K'-nearest landmarks of $r_{x_i}$.
Step 2: Search the K-nearest landmarks. As (2) of Figure 5 shows, we search the K-nearest landmarks of \( x_i \) among \( N_K(r_i^1) \) and denote them as \( N_K(x_i) \).

After the K-nearest landmarks approximation, we compute the similarity matrix \( B \) according to (17). For all data points, the complexity of step 1 is \( O(p^2(d + K)) \) and step 2 is \( O(NK(d + k)) \). The computational complexity of our similarity construction is \( O(p^2(d + K) + NK(d + k)) \). As \( k, K \ll p \ll n \), the dominant term in the complexity is \( O(NKd) \). Compared with the exact similarity construction of \( O(Npd) \) [4, 27], our method is much faster.

3.3. Bipartite Graph Partitioning

After obtaining the similarity matrix \( B \), we conduct graph partitioning on the graph Laplacian. The similarity matrix \( B \) reflects the relationships between \( X \) and \( R \). Therefore \( B \) can be interpreted as the cross-similarity matrix of the bipartite graph:

\[
G = \{X, R, B\}, \tag{18}
\]

where \( X \cup R \) is the node-set. As a result, the objective is changed to a bipartite graph partitioning problem.

We apply transfer cuts to efficiently compute the spectrum of graph Laplacian for spectral clustering. The \( k \) bottom eigenvectors \( u_1, u_2, \ldots, u_k \) for \( X \) side can be computed according to (7) and (8). Let \( U \in \mathbb{R}^{N \times k} \) be the matrix containing the vectors \( u_1, \ldots, u_k \) as columns, then matrix \( U \) will be the spectrum of graph Laplacian. In practice, we normalized \( U \) by its 1-norm as \( \tilde{U} \), then apply k-means clustering on \( \tilde{U} \) to obtain the final clustering results [17]. The k-means clustering is then performed on this embedding to obtain the \( k \) clusters as the final clustering result with \( O(Nk^2t) \) computational complexity. We summarize the divide-and-conquer based large-scale spectral clustering in Algorithm 3.

4. Discussion

4.1. Computational Complexity Analysis

In this section, we summarize the computational cost of the proposed method in each phase.

The divide-and-conquer based landmark selection takes \( O(Nad) \) time. The similarity construction takes \( O(NKd + p^2(d + K)) \) time. The eigen-decomposition takes \( O(NK(K + k) + p^3) \) time. The k-means discretization takes \( O(Nk^2t) \) time. With consideration to \( k, K, a \ll p \ll N \), the overall computational complexity of DnC-SC is \( O(N(ad + K^2 + Kk + Kd + k^2t) + p^3 + p^3(d + K)) \), where \( O(N(\alpha d)ad) \) is the dominant term. Table 1 provides a comparison of computational complexity of our DnC-SC algorithm against several other large-scale spectral clustering algorithms. The space complexity of DnC-SC is \( O(NK) \).

4.2. Relations with Other Methods

As a large-scale spectral clustering method, the proposed method is closely related to the methods in [4, 11]. We compare the proposed method with the two methods to discuss the improvements of the proposed method.

| Method   | R     | R     | R     |
|----------|-------|-------|-------|
| Nyström  | /     | \( O(Npd) \) | \( O(Np^p) \) |
| LSC-R    | /     | \( O(Npd) \) | \( O(Np^p) \) |
| LSC-K    | \( O(Npd) \) | \( O(Npd) \) | \( O(Np^p) \) |
| U-SPEC   | \( O(p^2dt) \) | \( O(Np^2d) \) | \( O(NK(K+k)+p^3) \) |
| DnC-SC   | \( O(Nad) \) | \( O(NKd) \) | \( O(NK(K+k)+p^3) \) |

* The final k-means is \( O(Nk^2t) \) for each method.

Firstly, we compare them on the landmark selection methods. Both the two methods [4, 11] directly or indirectly apply k-means based landmark selection. LSC-K method [4] directly conduct k-means algorithm to select landmarks within a high computational complexity \( O(Npd) \). While the U-SPEC [11] indirectly conduct k-means algorithm on a random set of samples to select landmarks, which finds a balance between k-means and random selection within \( O(p^2dt) \) time cost. Despite U-SPEC can efficiently find landmarks, it also has two limitations: 1) The quality of landmarks highly depends on how good the random set of samples is set up; 2) Since landmarks are not the centers for all data points, the center’s nature of landmark can not be used to approximate the similarity matrix. The proposed method uses the divide-and-conquer based landmark selection, which can effectively produce high-quality landmarks. We design an objection function (3) is to find the landmarks that best represent all data points with minimum RSS. We then propose
 Divide-and-conquer based Large-Scale Spectral Clustering

5. Experiments

In this section, we conduct experiments on five real and five synthetic datasets to evaluate the performance of the proposed DnC-SC methods. The comparison experiments against several state-of-the-art spectral clustering methods show better performance on clustering quality and efficiency for DnC-SC methods. Besides that, the analysis of the 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 thousand to as large as twenty million data points. Specifically, the five real datasets are USPS [6] 1, PenDigits [1] 2, Letters [10] 3, MNIST [5] 1, and Covertype [2] 4. The five synthetic datasets are Two Spiral-60K (TS-60K), Two Moons-1M (TM-1M), Three Circles-6M (TC-6M), Circles and Gaussians-10M (CG-10M) [11] 5, Flower-20M (FL-20M) [11] 5. Figure 6 shows the synthetic datasets. The properties of the datasets are summarized in Table 2.

We adopt two widely used evaluation metrics, i.e., Normalized Mutual Information (NMI) [22] and Accuracy (ACC) [26], to evaluate the clustering results. Let \( X = \{x_1, x_2, \ldots, x_n\} \) be the data matrix. For each data point \( x_i \), denote \( t_i \) and \( c_i \) as the cluster label of ground truth and obtained cluster label from clustering methods, respectively. The ACC is defined as:

\[
ACC = \frac{\sum_{i=1}^{n} \delta(t_i, \text{map}(c_i))}{n},
\]

where \( n \) is the number of data and \( \delta(t_i, c_i) \) is a function to check \( t_i \) and \( c_i \) are equal or not, returning 1 if equals otherwise returning 0. The \( \text{map}(c_i) \) is a best mapping function that maps each predicted label to the most possibly true cluster label by permuting operations [25].

Let \( T \) denote a set of clusters of ground truth and \( C \) obtained from clustering methods. Mutual information (MI) is defined as:

\[
MI(T, C) = \sum_{t_i \in T, c_i \in C} p(t_i, c_i) \ln \frac{p(t_i, c_i)}{p(t_i)p(c_i)},
\]

Table 2

| Dataset       | #Object | #Dimension | #Class |
|---------------|---------|------------|--------|
| USPS          | 9298    | 256        | 10     |
| PenDigits     | 10,992  | 16         | 10     |
| Letters       | 20,000  | 16         | 26     |
| MNIST         | 70,000  | 784        | 10     |
| Covertype     | 581,012 | 54         | 7      |
| TS-60K        | 600,000 | 2          | 3      |
| TM-1M         | 1,000,000 | 2   | 2        |
| TC-6M         | 6,000,000 | 2     | 3          |
| CG-10M        | 10,000,000 | 2    | 11         |
| FL-20M        | 20,000,000 | 2   | 13         |

(a) TS-60K (1%) (b) TM-1M (1%) (c) TC-6M (1%) (d) CG-10M (0.1%) (e) FL-20M (0.1%)

**Figure 6:** Illustration of the five synthetic datasets. Note that only a 1% or 0.1% samples of each dataset is plotted.

**References:**
1. [http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html](http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html)
2. [https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits](https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits)
3. [https://archive.ics.uci.edu/ml/datasets/Letter+Recognition](https://archive.ics.uci.edu/ml/datasets/Letter+Recognition)
4. [https://archive.ics.uci.edu/ml/datasets/covertype](https://archive.ics.uci.edu/ml/datasets/covertype)
5. [https://www.researchgate.net/publication/330706069](https://www.researchgate.net/publication/330706069)
where $p(t_i)$ and $p(c_i)$ are marginal probabilities that a sample happens to belong to cluster $t_i$ or $c_i$ while $p(t_i, c_i)$ is the joint probabilities that a sample happens to belong to cluster both $t_i$ and $c_i$. The NMI is the normalization of MI by the joint entropy as follow:

$$NMI(T, C) = \frac{\sum_{t_i \in T, c_i \in C} p(t_i, c_i) \ln \frac{p(t_i, c_i)}{p(t_i)p(c_i)}}{-\sum_{t_i \in T, c_i \in C} p(t_i, c_i) \ln (p(t_i, c_i))}, \quad (21)$$

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. Baseline Methods and Experimental Settings

In this experiment, we compare the proposed method with two baseline clustering methods, which are $k$-means clustering and spectral clustering (SC) [7], as well as six state-of-the-art large-scale spectral clustering methods. The compared spectral clustering methods are listed as follows:

1. **SC** [7]: original spectral clustering.
2. **Nyström** [9]: Nyström spectral clustering.
3. **LSC-K** [4]: landmark based spectral clustering using $k$-means based landmark selection.
4. **LSC-R** [4]: landmark based spectral clustering using random landmark selection.
5. **LSC-KH** [27]: Landmark-based spectral clustering using $k$-means partition to find the hubs as the landmarks.
6. **LSC-RH** [27]: Landmark-based spectral clustering using random partition to find the hubs as the landmarks.
7. **U-SPEC** [11]: Ultra-Scalable Spectral Clustering.

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

- We set the number of landmarks or representatives as $p = 1000$ for DnC-SC, U-SPEC, Nyström, LSC-K, and LSC-R methods. The parameter analysis on $p$ will be further conducted in Section 5.3.1.
- We set the $K = 5$ for the number of nearest neighbors for DnC-SC, U-SPEC, LSC-K, and LSC-R. The parameter analysis on $K$ will be further conducted in Section 5.3.2.
- The DnC-SC method has a unique parameter $\alpha$. In the experiments, $\alpha = 200$ is used for the datasets whose size is less than 100,000, otherwise $\alpha = 50$.

### 5.3. Comparison with Large-scale Spectral Clustering Methods

In this section, we compare the proposed DnC-SC method with five state-of-the-art spectral clustering methods, as well as the $k$-means clustering and original spectral clustering methods as the baseline methods.

We report the experimental results in Tables 3, 4 and 5, where we use N/A to denote the case when MATLAB reports the error of out of memory. Only two methods (proposed DnC-SC and U-SPEC) pass all datasets because they can approximately compute the similarity matrix within a limited memory. The proposed DnC-SC method achieves the best clustering performance of both ACC and NMI ten times on ten benchmark datasets according to Table 3 and 4. The proposed DnC-SC method achieves the best efficiency nine times on ten benchmark datasets according to Table 5.

In addition, we report the average performance score and rank for each method in Tables 3, 4 and 5. The proposed DnC-SC method achieves the best average scores of both ACC and NMI. The DnC-SC method shows average ranks of 1.50 of ACC and 1.40 of NMI, which implies the best clustering quality in all spectral clustering methods. Moreover, the DnC-SC method costs much less average time than the other competitors and achieves a rank of 1.50, which implies the most efficient method in this experiment. Overall, the proposed DnC-SC method shows significant effectiveness and efficiency comparing with six state-of-the-art large-scale spectral clustering methods.

We conduct a series of parameters analysis experiments to demonstrate the performance of the proposed method varying different parameter settings. We select four dataset (Letters, MNIST, TS-60K, and TM-1M) as benchmark datasets to conduct the following experiments.

#### 5.3.1. Number of Landmarks $p$

We first conduct parameter analysis to compare the large-scale spectral clustering methods by varying the number of landmarks $p$ (also called landmarks) and report the experimental results in Table 6. In general, we can see that a larger value of $p$ brings a better performance of ACC and NMI but cost more time. The proposed DnC-SC achieves the best ACC and NMI scores on all datasets except the MNIST. On MNIST dataset, the proposed DnC-SC method shows the second-best ACC and NMI scores after the LSC-K method. In terms of time cost, the proposed DnC-SC method shows the best efficiency on all datasets. Overall, the proposed DnC-SC method shows significant effectiveness and efficiency in this comparison.

#### 5.3.2. Number of Nearest Landmarks $K$

We then conduct parameter analysis to compare the large-scale spectral clustering methods by varying the number of the nearest landmark $K$ and report the experimental results in Table 7. Note that the Nyström method does not have the parameter $K$. Therefore, we do not show the results of the Nyström method in this experiment. According to Table 7, the performance of most methods varies for different $K$ values. The proposed method shows the best ACC and NMI of performance for the three of four datasets, and the second-best ACC and NMI on the MNIST dataset. Overall, the proposed DnC-SC shows superior effectiveness and the
best efficiency on this comparison.

5.3.3. Number of Nearest Landmarks \(K\) and selection rate \(\alpha\)

To further demonstrate the proposed method, we evaluate the performances by varying parameters \(K\) and \(\alpha\) and report the experimental results in Table 8. For proposed DnC-SC methods, the selection rate parameter \(\alpha\) directly affects the computational complexity of landmark selection, while the number of nearest landmarks \(K\) affects similarity construction, respectively. As we can see, a larger \(K\) or \(\alpha\) generally leads more time cost while not necessarily achieves better performance.
Divide-and-conquer based Large-Scale Spectral Clustering

Table 6
Clustering performance (ACC(%) and NMI(%) and time costs(s)) for different methods by varying number of landmark p.

| Dataset | Letters | MNIST | TS-60K | TM-1M |
|---------|---------|-------|--------|-------|
| ACC     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| NMI     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| Time cost | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |

Table 7
Clustering performance (ACC(%) and NMI(%) and time costs(s)) for different methods by varying number of nearest landmarks K.

| Dataset | Letters | MNIST | TS-60K | TM-1M |
|---------|---------|-------|--------|-------|
| ACC     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| NMI     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| Time cost | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |

* LSC-KH and LSC-RH cannot be conduct on the TM-60K and TM-1M dataset due to the memory bottleneck.

5.3.4. Efficiency analysis
To explore the efficiency of the proposed method in each computational phase, we report the time costs of three different phases: landmark selection, similarity construction, and graph partitioning. We choose LSC-K, LSC-R, and U-SPEC algorithms that have similar mechanisms for comparison. We list the strategies and methods used in each method in Table 11. The experimental results are reported in Table 12.

For landmark selection, the LSC-K and LSC-R methods apply k-means and a random selection, respectively; the U-SPEC method uses a hybrid selection that conducts k-means on a small set of random candidates; DnC-SC utilizes...
Divide-and-conquer based Large-Scale Spectral Clustering

Table 8
Clustering performance (ACC(%), NMI(%), and time costs(s)) for different methods by varying number of nearest landmark K and selection rate \( \alpha \).

| Dataset | Letters | MNIST | TS-60K | TM-1M |
|---------|---------|-------|--------|-------|
| ACC     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| NMI     | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| Time cost | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |

Table 9
Clustering performance (ACC(%), NMI(%), and time costs(s)) for DnC-SC using divide-and-conquer based landmark selection and \( k \)-means based landmark selection.

| Data     | Letters | MNIST | TS-60K | TM-1M |
|----------|---------|-------|--------|-------|
| ACC      | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| NMI      | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| Time cost | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |

Table 10
Clustering performance (ACC(%), NMI(%), and time costs(s)) for DnC-SC using approximate \( K \)-nearest landmarks and exact \( K \)-nearest landmarks.

| Data     | Letters | MNIST | TS-60K | TM-1M |
|----------|---------|-------|--------|-------|
| ACC      | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| NMI      | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |
| Time cost | ![Graph](image) | ![Graph](image) | ![Graph](image) | ![Graph](image) |

the divide-and-conquer selection. Looking at the runtime of the landmark selection, we see that the random selection of LSC-R takes a little time, while the \( k \)-means selection takes much more time. The divide-and-conquer selection of DnC-SC is the second-fastest method just behind the random selection.

For similarity construction, the LSC-K and LSC-R compute the exact similarity matrix without approximation, while U-SPEC and DnC-SC calculate the similarity by approximate schemes. Compared with U-SPEC, DnC-SC uses the results of landmark selection to improve the approximate scheme. For the runtime of similarity construction, we find that DnC-SC takes significantly less time than other methods, especially for the larger-scale dataset (TM-1M). Note that the approximate similarity matrix of U-SPEC takes more time than LSC-K or LSC-R in MNIST dataset. However, the similarity of U-SPEC takes less computational complexity than LSC-K or LSC-R. This is because U-SPEC uses serial calculations in the approximation process. In MATLAB, it will be much faster to perform the approximation in a batch processing manner (with optimized matrix computation) than in a serial processing manner.
Table 11
Comparison for three phases for different methods.

| Phase                      | LSC-K | LSC-R | U-SPEC | DnC-SC |
|----------------------------|-------|-------|--------|--------|
| Landmark Selection         | k-means | Random | Hybrid representative selection | Divide-and-conquer selection |
| Similarity Construction    | Exact  | Exact  | Approximate | Approximate |
| Graph Partitioning         | SVD based | SVD based | Transfer cuts | Transfer cuts |

Table 12
Comparison of time costs in each phase for different methods.

For graph partitioning, LSC-K and LSC-R utilize SVD based method, while U-SPEC and DnC-SC apply transfer cuts. Theoretically, both graph partitioning methods can be considered as efficient solutions for bipartite graph partitioning [15, 4]. But the transfer cuts take less computational complexity. In Table 12, we can see that U-SPEC and DnC-SC take less time than LSC-K and LSC-R, which is consistent with the theoretical complexity.

Overall, DnC-SC shows the best efficiency in four methods, which is mainly due to the proposed landmark selection and approximate similarity construction.

5.4. Influence of Landmark Selection Strategies

Some existing works have shown that the performance of large-scale spectral clustering heavily relies on the proper strategy of landmark selection [12]. In our proposed landmark selection, we propose a divide-and-conquer selection strategy and light-k-means to find a good balance between effectiveness and efficiency. We test the purposed method with different landmark selection methods, i.e., k-means based landmark selection, divide-and-conquer selection without light-k-means, and divide-and-conquer selection with light-k-means.

In this section, we compare the performances between the divide-and-conquer based landmark selection and the k-means based landmark selection. The experimental results are reported in Table 9. As we mentioned, the divide-and-conquer based landmark selection algorithm recursively solves the optimization problems 3, which k-means methods can also solve. We have pointed out the lack of efficiency of directly applying k-means on large-scale datasets in Section 4.1. Note that the number of maximum iterations of k-means in landmark selection is turned as 5, which is the same setting as LSC-K and U-SPEC implementation. In Table 9, k-means based landmark selection algorithm generally shows better ACC and NMI on most datasets except TM-1M dataset, while the difference in performance is not significant. Compared to k-means based selection, our divide-and-conquer based landmark selection algorithm strikes a balance between efficiency and effectiveness. It achieves significantly better efficiency than the k-means based selection and yields competitive clustering quality compared to the k-means based selection.

5.4.1. Performance comparison on simulation scenarios

To further investigate the performance of divide-and-conquer selection, we conduct a simulation experiment to simulate different scenarios for landmark selection. For landmark selection, the number of landmarks is considered much larger than the desired number of clusters. If we view the landmark selection as a clustering task, then the landmark selection will be considered as a special clustering case with a large number of clusters. Therefore, we generate four synthetic datasets with 500, 1000, 1500, 2000 clusters, respectively. The synthetic datasets are 2-dimensional isotropic Gaussian blobs, which are shown in Figure 7. We treat divide-and-conquer selection as a clustering algorithm to compare the clustering performance with k-means. We report the clustering performance of NMI and time costs for all simulation scenarios in Table 13 and Table 14.

Though divide-and-conquer selection shows slightly lower NMI than k-means, its time cost is much less. As landmark increases, the performance degradation associated with divide-and-conquer selection becomes progressively insignificant, while the improvement of efficiency becomes more significant. The experimental results imply that the divide-and-conquer selection is suitable for a larger number of land-
marks while $k$-means selection is suitable for a smaller number of landmarks. Usually, more landmarks will lead to a better clustering result for large-scale spectral clustering [4, 11]. Thus, the divide-and-conquer selection is more suitable than $k$-means selection for large-scale spectral clustering.

5.5. Influence of Approximated $K$-nearest Landmarks

In this section, we compare the approximated $K$-nearest landmarks and exact $K$-nearest landmarks. The experimental results are reported in Table 10. The approximated $K$-nearest landmarks approach first finds the possible candidates according to the center’s nature of landmarks and then searches the $K$-nearest landmarks among them. The exact $K$-nearest landmarks approach costs $O(Npd)$ computational time, while the proposed approximation can reduce the time cost to $O(NKd)$. As the Tables 10 shows, the exact $K$-nearest landmarks approach achieves slightly better ACC and NMI scores than the proposed approximation. However, the performances of the two methods are not significantly different. In terms of time cost, the proposed approximation approach shows highly efficient performance compared with the exact $K$-nearest landmarks. Note that the exact $K$-nearest landmarks approach can not be conducted on datasets whose sizes are more than one million due to the high computational cost. Overall, the proposed approximate $K$-nearest landmark approach shows the robustness and efficiency of this experiment.

5.6. Ablation Study

To strike a good balance between efficiency and effectiveness, the proposed method applies three strategies: (a) divide-and-conquer selection, (b) light-$k$-means, and (c) approximate of $K$-nearest landmarks. An ablation study about the influence of the combination of each part is conducted to show the contribution of each strategy. The experimental results are reported in Table 15. Modules (a) and (b) are used in landmark selection. In Table 15, DnC-$k$-means indicates a modified divide-and-conquer selection that utilizes $k$-means algorithm for the dividing process, and DnC-light-$k$-mean indicates the original divide-and-conquer selection that utilizes the light-$k$-means algorithm for the dividing process. To show the effects of (a) and (b), we choose $k$-means selection as the baseline. For $K$-nearest landmarks, we provide the exact $K$-nearest landmark option for each landmark selection method. There are three landmark selections and two $K$-nearest landmark methods provided in this ablation study. Thus we have six combinations for comparison.

Table 13 shows the performance on six combinations according to different landmark selection and $K$-nearest landmarks methods. The bold texts represent the best ACC, NMI, and Time for each dataset. We first compare the different landmark selection methods: the $k$-means selection archives the best ACC on two datasets and the best NMI on three datasets, but takes much more runtime on all datasets; DnC-$k$-means selection archives the best ACC on two datasets and best NMI on one dataset with much less runtime than $k$-means selection; our DnC-light-$k$-means takes the least time on all datasets and shows a competitive performance of ACC and NMI. For $K$-nearest landmark, the extra approach archives the best ACC on three datasets and the best NMI on two datasets, which slightly outperforms the approximate approach. We also report the average score for each combination. The combination of $k$-means and extra $K$-nearest landmarks show the best average scores of ACC and NMI, but the most time-consuming. Our proposed method that is the combination of DnC-light-$k$-means and approximate $K$-nearest landmarks shows the fastest speed and competitive performance of ACC and NMI.

Overall, the proposed method significantly improves the efficiency of large-scale spectral clustering while keeping the clustering quality acceptable. In detail, we can see that the use of strategy (a) provides the most important contribution to the computational efficiency, while modules (b) and
Table 15: Ablation Study on the proposed divide-and-conquer selection strategy, light-\(k\)-means, and approximate of \(K\)-nearest landmarks.

| Datasets   | Landmark selection | \(K\)-nearest landmarks | Approximate | Exact |
|------------|-------------------|--------------------------|-------------|------|
|            |                   | ACC(%)                  | NMI(%)      | Time(s) | ACC(%) | NMI(%) | Time(s) |
| Letters    | \(k\)-means      | 34.06                    | 46.58       | 3.89    | 34.41  | 45.56  | 4.05    |
|           | DnC-\(k\)-means  | 34.21                    | 45.19       | 1.22    | 33.76  | 45.17  | 1.34    |
|           | DnC-light-\(k\)-means | 33.54           | 45.37       | 0.90    | 33.93  | 45.91  | 1.05    |
| MNIST      | \(k\)-means      | 75.34                    | 73.07       | 15.29   | 79.28  | 74.74  | 29.02   |
|           | DnC-\(k\)-means  | 74.46                    | 73.11       | 9.50    | 74.12  | 74.12  | 24.81   |
|           | DnC-light-\(k\)-means | 74.24           | 72.00       | 5.11    | 74.04  | 74.04  | 21.00   |
| TS-60K     | \(k\)-means      | 83.27                    | 77.18       | 165.21  | 86.41  | 76.51  | 172.72  |
|           | DnC-\(k\)-means  | 81.06                    | 73.92       | 8.14    | 84.30  | 70.15  | 12.75   |
|           | DnC-light-\(k\)-means | 81.00           | 73.84       | 4.01    | 80.82  | 73.12  | 9.12    |
| TM-1M      | \(k\)-means      | 99.23                    | 99.50       | 3997.12 | 99.95  | 99.59  | 4023.12 |
|           | DnC-\(k\)-means  | 99.95                    | 99.48       | 12.78   | 99.97  | 99.57  | 25.65   |
|           | DnC-light-\(k\)-means | 99.96           | 99.52       | 6.46    | 99.95  | 99.45  | 19.30   |
| Avg. score | \(k\)-means      | 72.98                    | 74.08       | 1045.38 | 75.01  | 74.10  | 1057.23 |
|           | DnC-\(k\)-means  | 72.55                    | 72.93       | 7.91    | 73.04  | 72.25  | 16.14   |
|           | DnC-light-\(k\)-means | 72.19           | 72.68       | 4.12    | 72.19  | 73.13  | 12.62   |

(c) further reduce the computational cost.

6. Conclusion
In this paper, we propose a large-scale clustering method, termed divide-and-conquer based spectral clustering (DnC-SC). In DnC-SC, a divide-and-conquer based landmark selection algorithm is designed to obtain the landmarks effectively. A new approximate similarity matrix construction approach is proposed to utilize the center's nature of the landmarks to fast construct the similarity matrix between data points and \(K\)-nearest landmarks. Finally, the bipartite graph partition is conducted to obtain the final clustering results. The experimental results on synthetic and real-world datasets show that the proposed method outperforms other state-of-the-art large-scale spectral clustering methods.

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

References
[1] Asuncion, A., Newman, D., 2007. UCI machine learning repository.
[2] Blackard, J.A., Dean, D.J., 1999. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Computers and electronics in agriculture 24, 131–151.
[3] Bouneffouf, D., Biroli, L., 2015. Sampling with minimum sum of squared similarities for nystrom-based large scale spectral clustering. in: Twenty-Fourth International Joint Conference on Artificial Intelligence.
[4] Cai, D., Chen, X., 2014. Large scale spectral clustering via landmark-based sparse representation. IEEE transactions on cybernetics 45, 1669–1680.
[5] Cai, D., He, X., Han, J., 2011. Speed up kernel discriminant analysis. The VLDB Journal 20, 21–33.
[6] Cai, D., He, X., Han, J., Huang, T.S., 2010. Graph regularized non-negative matrix factorization for data representation. IEEE transactions on pattern analysis and machine intelligence 33, 1548–1560.
[7] Chen, W.Y., Song, Y., Bai, H., Lin, C.J., Chang, E.Y., 2008. A survey of kernel and spectral methods for clustering. Pattern recognition 41, 176–190.
[8] Filippone, M., Camastra, F., Masulli, F., Rovetta, S., 2008. A survey of kernel and spectral methods for clustering. Pattern recognition 41, 176–190.
[9] Fowlkes, C., Belongie, S., Chung, F., Malik, J., 2004. Spectral grouping using the nystrom method. IEEE transactions on pattern analysis and machine intelligence 26, 214–225.
[10] Frey, P.W., Slate, D.J., 1991. Letter recognition using holland-style adaptive classifiers. Machine learning 6, 161–182.
[11] Huang, D., Wang, C.D., Wu, J.S., Lai, J.H., Kwoh, C.K., 2019. Ultra-scalable spectral clustering and ensemble clustering. IEEE Transactions on Knowledge and Data Engineering 32, 1212–1226.
[12] Li, H., Ye, X., Imakura, A., Sakurai, T., 2020. Hubness-based sampling method for nystrom spectral clustering, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE. pp. 1–8.
[13] Li, Y., Nie, F., Huang, H., Huang, J., 2015. Large-scale multi-view spectral clustering via bipartite graph, in: Proceedings of the AAAI Conference on Artificial Intelligence.
[14] Li, Z., Chen, J., 2015. Superpixel segmentation using linear spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1356–1363.
[15] Li, Z., Wu, X.M., Chang, S.F., 2012. Segmentation using superpixels: A bipartite graph partitioning approach, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE. pp. 789–796.
[16] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., Wu, S., 2013. Understanding and enhancement of internal clustering validation measures. IEEE transactions on cybernetics 43, 982–994.
[17] Ng, A.Y., Jordan, M.I., Weiss, Y., 2002. On spectral clustering: Anal-
yis and an algorithm, in: Advances in neural information processing systems, pp. 849–856.

[18] Pentney, W., Meila, M., 2005. Spectral clustering of biological sequence data, in: AAAI, pp. 845–850.

[19] Rafailidis, D., Constantinou, E., Manolopoulos, Y., 2017. Landmark selection for spectral clustering based on weighted pagerank. Future Generation Computer Systems 68, 465–472.

[20] Rokach, L., Maimon, O., 2005. Clustering methods, in: Data mining and knowledge discovery handbook. Springer, pp. 321–352.

[21] Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence 22, 888–905.

[22] Slonim, N., Tishby, N., 2000. Agglomerative information bottleneck, in: Advances in neural information processing systems, pp. 617–623.

[23] Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and computing 17, 395–416.

[24] Xu, R., Wunsch, D., 2008. Clustering. volume 10. John Wiley & Sons.

[25] Xu, W., Liu, X., Gong, Y., 2003. Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 267–273.

[26] Yan, D., Huang, L., Jordan, M.I., 2009. Fast approximate spectral clustering, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 907–916.

[27] Ye, X., Li, H., Sakurai, T., Liu, Z., 2018. Large scale spectral clustering using sparse representation based on hubness, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE. pp. 1731–1737.

[28] Ye, X., Sakurai, T., 2018. Spectral clustering with adaptive similarity measure in kernel space. Intelligent Data Analysis 22, 751–765.

[29] Zha, H., He, X., Ding, C., Simon, H., Gu, M., 2001. Bipartite graph partitioning and data clustering, in: Proceedings of the tenth international conference on Information and knowledge management, pp. 25–32.

[30] Zhang, K., Tsang, I.W., Kwok, J.T., 2008a. Improved nyström low-rank approximation and error analysis, in: Proceedings of the 25th international conference on Machine learning, pp. 1232–1239.

[31] Zhang, X., Jiao, L., Liu, F., Bo, L., Gong, M., 2008b. Spectral clustering ensemble applied to sar image segmentation. IEEE Transactions on Geoscience and Remote Sensing 46, 2126–2136.

[32] Zhang, X., Zong, L., You, Q., Yong, X., 2016. Sampling for nyström extension-based spectral clustering: incremental perspective and novel analysis. ACM Transactions on Knowledge Discovery from Data (TKDD) 11, 1–25.

[33] Zhang, Z., Kulkarni, S.R., 2014. Detection of shilling attacks in recommender systems via spectral clustering, in: 17th International Conference on Information Fusion (FUSION), IEEE. pp. 1–8.