

Research Article

Large-Scale Spectral Clustering Based on Representative Points

Libo Yang ¹, Xuemei Liu ¹, Feiping Nie ², and Mingtang Liu ¹

¹School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, Henan, China

²School of Computer Science, Center for OPTIMAL, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

Correspondence should be addressed to Xuemei Liu; liuxuemei@ncwu.edu.cn and Feiping Nie; feipingnie@gmail.com

Received 8 September 2019; Accepted 12 November 2019; Published 9 December 2019

Academic Editor: Rafał Stanisławski

Copyright © 2019 Libo Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Spectral clustering (SC) has attracted more and more attention due to its effectiveness in machine learning. However, most traditional spectral clustering methods still face challenges in the successful application of large-scale spectral clustering problems mainly due to their high computational complexity $o(n^3)$, where n is the number of samples. In order to achieve fast spectral clustering, we propose a novel approach, called representative point-based spectral clustering (RPSC), to efficiently deal with the large-scale spectral clustering problem. The proposed method first generates two-layer representative points successively by BKHK (balanced k-means-based hierarchical k-means). Then it constructs the hierarchical bipartite graph and performs spectral analysis on the graph. Specifically, we construct the similarity matrix using the parameter-free neighbor assignment method, which avoids the need to tune the extra parameters. Furthermore, we perform the coclustering on the final similarity matrix. The coclustering mechanism takes advantage of the cooccurring cluster structure among the representative points and the original data to strengthen the clustering performance. As a result, the computational complexity can be significantly reduced and the clustering accuracy can be improved. Extensive experiments on several large-scale data sets show the effectiveness, efficiency, and stability of the proposed method.

1. Introduction

Clustering is one of the fundamental topics in unsupervised learning. It has been widely and successfully applied in data mining, pattern recognition, and many other fields. Spectral clustering is one of the most popular methods used in unsupervised clustering tasks [1–4]. Especially, it performs well in nonconvex pattern and linear nonseparable clusters and converges to the global optimal solution [5]. However, spectral clustering is limited in its applicability to large-scale problems. Its bottleneck is the high computational complexity [6–8]. Many approaches have been proposed to speed up spectral clustering [9–11]. Unfortunately, these methods usually sacrifice a lot of information of the raw data, resulting in performance degradation.

The traditional spectral clustering needs two independent steps: constructing similarity graph and performing spectral analysis [12]. Both the steps are computational expensive for large-scale data, and their

computational complexity is $o(n^2)$ and $o(n^3)$, respectively. The conventional spectral clustering has three methods to construct the similarity graph which is constructed by pairwise similarities or pairwise distances. The goal is to model the local neighborhood relationships between the data points [13]. The first method is to construct the ε -neighborhood graph in which ε is the pairwise distance. All points whose pairwise distances are smaller than ε can be connected. In this method, a large amount of information between sample points is discarded because of this single and rough criterion. The second method is to generate similarity graph by the k-nearest neighbor method in which two vertices p and q are connected if the distance between p and q is among the k -th smallest distances from p to other objects. The last similarity graph is the fully connected graph. That means all points with positive similarity are connected with each other. The computational cost of all graphs mentioned above is very expensive. In the past years, a lot of extensions of spectral clustering have been proposed. Wang

and Wu proposed to simultaneously learn clustered orthogonal projection and optimized local graph structure for each view, while enjoying the same magnitude over them both for all views, leading to a superior multiview spectral clustering consensus [14]. Houthuys et al. proposed a new model multiview kernel spectral clustering (MVKSC). This model is formulated as a weighted kernel canonical correlation analysis in a primal-dual optimization setting typical of least-squares support-vector machines (LSSVM). A coupling term enforces the clustering scores corresponding to the different views to align [15]. Tao et al. proposed a novel robust spectral ensemble clustering algorithm. It learns low-rank representation for the coassociation matrix to uncover the cluster structure and handle the noises, and meanwhile, it performs spectral clustering with the learned representation to seek for a consensus partition [16]. Huang et al. proposed ultrascaleable spectral clustering (U-SPEC) and ultrascaleable ensemble clustering (U-SENC). In U-SPEC, a hybrid representative selection strategy and a fast approximation method for K-nearest representatives are proposed for the construction of a sparse affinity submatrix. In U-SENC, multiple U-SPEC clusters are further integrated into an ensemble clustering framework to enhance the robustness of U-SPEC [17]. Yang et al. designed a constraint projection for a semisupervised spectral clustering ensemble (CPSSSCE) model. In this method, the original data are transformed to lower-dimensional representations by constraint projection before base clustering. Then, a similarity matrix is constructed using the base clustering results and modified using pairwise constraints. Finally, the spectral clustering algorithm is applied to process the similarity matrix to obtain a consensus cluster result [18].

In recent years, many spectral clustering methods for large-scale data have been proposed. Zhao et al. proposed a spectral clustering based on iterative optimization (SCIO), which solves the spectral decomposition problem of large-scale and high-dimensional data set, and this method performs on multitask clustering [19]. The nonnegative matrix factorization (NMF) has been proposed as the relaxation technique for clustering with excellent performance [20, 21]. Yang et al. proposed a multitask spectral clustering model (MTSC) by exploring two types of correlations: intertask clustering correlation and intratask learning correlation [22]. He et al. reduced the complexity of spectral clustering by employing random Fourier features to explicitly represent data in kernel space [23]. Semertzidis et al. proposed an efficient spectral clustering method for large-scale data sets in which a set of pairwise constraints were given to increase clustering accuracy and reduce clustering complexity [24]. These methods greatly extend spectral clustering for large-scale data.

Recently, the representative point-based graph has widely been adopted in the spectral-based method to speed up the procedure. Shinnou and Sasaki [25] constructed the similarity matrix on “committees” of raw data points, and large-scale spectral clustering was performed using the reduced similarity matrix. Yan et al. [26] introduced a fast

spectral clustering method which used the k-means method to produce a set of reduced representative points for approximate spectral clustering (KASP: k-means-based approximate spectral clustering). Liu et al. [27] addressed the scalability issue plaguing graph-based semisupervised learning via the anchor points. Liu et al. [28] proposed an efficient cluster algorithm for large-scale graph data using spectral methods. In order to compress the original graph into a sparse bipartite graph, they repeatedly generate a small number of “supernodes” connected to the regular nodes. Cai and Chen [29] and Cai [30] proposed landmark-based spectral clustering and spectral dimensionality reduction. Li et al. [31] adopted the salient point-based subbipartite graph to achieve large-scale multiview spectral clustering. Wang et al. [32] explored the spectral and spatial properties of the hyperspectral image and proposed a novel method, called fast spectral clustering with anchor graph (FSCAG). This method solved the large-scale hyperspectral image clustering problem. These methods mentioned above adopt representative point-based strategy to construct the similarity graph to accelerate the procedure of spectral clustering.

The representative point generation is one of the extremely important steps in representative point-based spectral clustering methods. These generated points directly affect the final performance. There are two main strategies to generate representative points. One is random selection, and the other is the k-means method. The random generation strategy is efficient, but the performance is not stable because of the randomness of representative points. Generally speaking, the k-means generation strategy can achieve good performance while the computational cost is expensive. Much effort has been devoted for accelerating the k-means procedure, e.g., early stopping iteration [33] or performing down-sampling on data, but they can also sacrifice some performance.

To tackle the problem, a novel and efficient representative point-based spectral clustering method is proposed to deal with large-scale data sets. Three main contributions of this paper are listed as follows:

- (1) The two-layer bipartite graph is constructed using the generated representation points by BKHK. BKHK has low computational complexity and high performance compared with k-means.
- (2) We construct the similarity matrix between adjacent layers using the parameter-free neighbor assignment method, which avoids extra parameters. Furthermore, the final similarity matrix is easily obtained by multiplying the similarity matrix between adjacent layers.
- (3) We perform the coclustering on the final similarity matrix. The coclustering mechanism takes advantage of the cooccurring cluster structure among the representative points and the original data to strengthen the clustering performance.
- (4) Extensive experiments on several large-scale data sets demonstrate the effectiveness, efficiency, and stability of the proposed method.

2. Representative Points Generation

The most important step of representative point-based spectral clustering is the generation of representative points. In this paper, we design two-layer representative points by BKHK to gradually reduce the data size. BKHK adopts balanced binary tree structure; in other words, it iteratively segments the data into two cluster with the same number of samples [34].

Given a data matrix $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$, where $x_i \in \mathbb{R}^d$ denotes the i -th sample, n is the number of samples, and d is the dimensionality of the data. The two class k-means can be formulated as follows:

$$\min_{G \in \text{Ind}, \mathbf{1}^T G = [\kappa, \iota]} \|X - GC^T\|_F^2, \quad (1)$$

where $G \in \mathbb{R}^{n \times 2}$ is the index matrix, g_{ij} equals 1 or 0, the value of g_{i1} is 1, and the value of g_{i2} is 0 if the i -th sample belongs to the first cluster, or the value of g_{i1} is 0 and the value of g_{i2} is 1 otherwise, $C \in \mathbb{R}^{d \times 2}$ is the center of the cluster, $\mathbf{1}$ is the column-vector of all ones, κ and ι are the number of samples in these two clusters, and k is the integer portion of $n/2$. Therefore, we have $l = n - k$. Furthermore, problem (1) can be rewritten as

$$\min_{G \in \text{Ind}, \mathbf{1}^T G = [\kappa, \iota]} \sum_{i=1}^n \sum_{j=1}^2 \|x_i - c_j\|_2^2 g_{ij}, \quad (2)$$

where c_j is the j -th column of C .

We define matrix $E \in \mathbb{R}^{n \times 2}$ and the j -th element of the i -th row of E is $e_{ij} = \|x_i - c_j\|_2^2$, so problem (2) can be rewritten as

$$\min_{G \in \text{Ind}, \mathbf{1}^T G = [\kappa, \iota]} \text{Tr}(E^T G). \quad (3)$$

For convenience, let g denote the first column of G , obviously, the second column is $1 - g$. Substitute g and $1 - g$ into problem (3), we have

$$\min_{g \in \{0,1\}, \mathbf{1}^T g = \kappa} g^T (e_1 - e_2). \quad (4)$$

Let $g_i = 1$ when the i -th element of $e_1 - e_2$ is the κ minimum of all its elements. Obviously, the solution to the problem in problem (4) is obtained.

We can obtain the first-layer representative points by performing above process iteratively. Then the procedure is repeated on the first-layer representative points to generate the second-layer representative points.

3. Similarity Matrix

Similar to conventional similarity graph construction, the similarity graph construction between the obtained representative points and raw points also has the problem of selecting the neighbor assignment strategy. The kernel-based neighbor assignment strategy usually is used in conventional methods, but it always brings extra parameters [13]. A parameter-free method is adopted in this paper [35]. Let $U \in \mathbb{R}^{m \times d}$ denote the generated representative points, and $U_{\langle i \rangle}$ is the set of k -nearest representative points for the i -th

sample. $h(x_i, u_j) = \|x_i - u_j\|_2^2$ is the distance between the i -th sample and its j -th nearest representative point. The neighbor assignment strategy can be formulated as follows:

$$\min_{b_i^j \mathbf{1} = 1, b_i \geq 0} \sum_{j=1}^m h(x_i, u_j) b_{ij} + \gamma \sum_{j=1}^m b_{ij}^2, \quad (5)$$

where $B \in \mathbb{R}^{n \times m}$ is the similarity matrix between raw data and representative points and b_{ij} is the element of B . b_{ij} denotes the similarity between the i -th data point and the j -th representative point. Follow Nie et al. [35], γ is equal to $(k/2)h(i, k+1) - (1/2)\sum_{j=1}^k h(i, j)$. The solution to problem (5) is as follows:

$$b_{ij} = \frac{h(x_i, u_{k+1}) - h(x_i, u_j)}{\sum_{j'=1}^k (h(x_i, u_{k+1}) - h(x_i, u_{j'}))}. \quad (6)$$

The two-layer representative points are successively generated by performing BKHK. $B_1 \in \mathbb{R}^{n \times m_1}$ denotes the similarity matrix between original data points and the first-layer representative points which can be obtained by solving problem (6), and $B_1 \mathbf{1}$ equals $\mathbf{1}$. $B_2 \in \mathbb{R}^{m_1 \times m_2}$ denotes the similarity matrix between the first-layer representative points and the second-layer representative points which can be obtained by the same process, and $B_2 \mathbf{1}$ equals $\mathbf{1}$. The final similarity matrix between the raw data points and the second-layer representative points can be obtained as follows:

$$A = B_1 \times B_2, \quad (7)$$

where $A \in \mathbb{R}^{n \times m_2}$ is the final similarity matrix.

4. Coclustering on Similarity Matrix

As the same as document data, duality exists between the raw data points and the representative points. The representative points can be clustered based on their relations with the corresponding raw data clusters, while the raw data clusters are obtained according to their associations with distinct representative point clusters. In order to make full use of the duality information and strengthen the clustering performance, the coclustering method is adopted on the similarity matrix between the raw data points and the second-layer representative points.

4.1. Graph Partitioning. A signifies association between an original point, and a representative point signifies an edge in bipartite graph. It is easy to verify that the adjacency matrix of the bipartite graph can be written as follows:

$$M = \begin{bmatrix} \mathbf{0} & A \\ A^T & \mathbf{0} \end{bmatrix}, \quad (8)$$

where $M \in \mathbb{R}^{(n+m_2) \times (n+m_2)}$ denotes the adjacency matrix and $A \in \mathbb{R}^{n \times m_2}$ is the similarity matrix between the raw data points and the second-layer representative points.

Therefore, the degree matrix D of M can be obtained as follows:

$$D = \begin{bmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix}, \quad (9)$$

where D_1 and D_2 are the diagonal matrices whose entries are $D_1(i, i) = \sum_j A_{ij}$ and $D_2(j, j) = \sum_i A_{ij}$, respectively. Obviously, the Laplacian matrix is

$$L = D - M = \begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix}. \quad (10)$$

To partition the bipartite graph, the optimization problem of finding the minimum normalized cut can be formalized as the second eigenvector of the generalized eigenvalue problem with suitable relaxation [1]:

$$Lx = \lambda Dx. \quad (11)$$

Substitute equations (9) and (10) into equation (11), we get

$$\begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (12)$$

Furthermore, equation (12) can be written as follows:

$$\begin{aligned} D_1^{1/2} x_1 - D_1^{-1/2} A x_2 &= \lambda D_1^{1/2} x_1, \\ -D_2^{-1/2} A^T x_1 + D_2^{1/2} x_2 &= \lambda D_2^{1/2} x_2. \end{aligned} \quad (13)$$

Letting $u = D_1^{1/2} x_1$ and $v = D_2^{1/2} x_2$, and after a little algebraic manipulation, equation (13) can be written as follows:

$$\begin{aligned} D_1^{-1/2} A D_2^{-1/2} v &= (1 - \lambda) u, \\ D_2^{-1/2} A^T D_1^{-1/2} u &= (1 - \lambda) v. \end{aligned} \quad (14)$$

Obviously, equation (14) is the singular value decomposition of the normalized matrix $A_n = D_1^{-1/2} A D_2^{-1/2}$. u is the left singular vector, and v is the right singular vector while $(1 - \lambda)$ is the corresponding singular value. Thus instead of directly computing the eigenvector of (12), we prefer to compute the left and right singular vectors of A_n to speed up the algorithm.

In order to achieve multipartitioning, the data set can be formed as follows:

$$Z = \begin{bmatrix} D_1^{-1/2} U \\ D_2^{-1/2} V \end{bmatrix}, \quad (15)$$

where $U = [u_2, u_3, \dots, u_{k+1}]$ and $V = [v_2, v_3, \dots, v_{k+1}]$ come from the $k + 1$ minimum left and right singular vectors of A_n .

To approximate the optimal multipartitioning, we can look for points m_1, m_2, \dots, m_k such that the sum of squares is minimized [27]:

$$\sum_{j=1}^k \sum_{Z(i) \in m_j} \|Z(i) - m_j\|^2. \quad (16)$$

Obviously, the minimum of (16) can be achieved by the classical k-means algorithm.

4.2. Representative Point-Based Spectral Clustering. As mentioned, two-layer representative points can be successively generated by BKHK for large-scale spectral clustering and similarity graph can be constructed. Then coclustering can be performed on the final similarity matrix between original data points and representative points. So we propose a novel spectral clustering approach for large-scale data, called representative point-based spectral clustering (RPSC). The detailed algorithm is summarized in Algorithm 1.

5. Experiments

In this section, several experiments are conducted to demonstrate the effectiveness and efficiency of the proposed RPSC.

5.1. Data Sets Description. In the experiments, four large-scale data sets are collected to illustrate the performance of different clustering methods. These data sets include two handwritten digit data sets, USPS, MNIST, one connect-4 game data set, CONNECT-4, and one capital letters in the English alphabet, LETTER. These data sets were downloaded from the UCI machine learning repository and the LibSVM data sets page. The brief description is listed below:

USPS. A data set of handwritten digits (0–9). It contains 9298 samples from ten classes. Each sample has 256 features.

MNIST. A data set of handwritten digits (0–9). It has 70000 samples from ten classes. Each sample has 784 features.

CONNECT-4. A data set of connect-4 game. It consists of 67557 samples from three classes. Each sample has 126 features.

LETTER. A data set of 26 capital letters in the English alphabet. It is composed of 20000 samples from 26 classes. Each sample has 16 features.

5.2. Evaluation Metric. All the codes in the experiments are implemented in MATLAB R2016a and run on a Windows 8.1 machine with 32 GB main memory. Every method was run 10 times, and the mean results were recorded. The average clustering accuracy and the average clustering time are displayed in Tables 1 and 2, respectively. The clustering accuracy and the clustering time on different number of two-layer representative points of two data sets are recorded in Figures 1–4.

5.3. Experimental Results. The performance of five methods evaluated by accuracy is reported in Table 1. The conventional algorithm SC can only run on the small data sets USPS and Letter, and the clustering accuracy is low. LSC-K achieves high clustering accuracy on USPS and MNIST, but it does not perform well on LETTER and CONNECT-4. LSC-R has the low clustering accuracy on almost all data sets. KASP only performs well on one of the four data sets. The proposed RPSC achieves pretty high performance on all

Input: Data matrix
 $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$;
 Output: k clusters;
 (1) Produce the two-layer representative points using BKHK successively;
 (2) Construct the similarity matrixes B1 and B2 according to (6);
 (3) Construct the final similarity matrix A between original data and the second-layer representative points according to (7);
 (4) Compute D1 and D2: $D_1(i, i) = \sum_j A_{ij}$; $D_2(j, j) = \sum_i A_{ij}$;
 (5) Compute $A_n = D_1^{-1/2} A D_2^{-1/2}$;
 (6) Compute k singular vectors of A_n : $u_2, u_3, \dots, u_{\ell+1}, v_2, v_3, \dots, v_{\ell+1}$;
 (7) Form the matrix $Z = \begin{bmatrix} D_1^{-1/2} U \\ D_2^{-1/2} V \end{bmatrix}$;
 (8) Run the k-means algorithm on the matrix Z to obtain the desired k -way multipartitioning.

ALGORITHM 1: Representative point-based spectral clustering.

TABLE 1: Clustering accuracy.

Data sets	Spectral clustering methods				
	SC	LSC-K	LSC-R	KASP	RPSC
USPS	43.27	60.83	58.30	55.71	61.65
MNIST	—	50.40	41.85	44.56	53.53
Connect-4	—	24.96	26.42	36.90	50.38
Letter	23.61	23.24	21.73	16.65	26.98

TABLE 2: Clustering time.

Data sets	Spectral clustering methods				
	SC	LSC-K	LSC-R	KASP	RPSC
USPS	430.35	186.25	2.28	2.55	23.26
MNIST	—	432.40	8.17	109.06	49.93
Connect-4	—	149.29	5.07	21.68	20.93
Letter	635.86	27.85	4.45	1.40	14.25

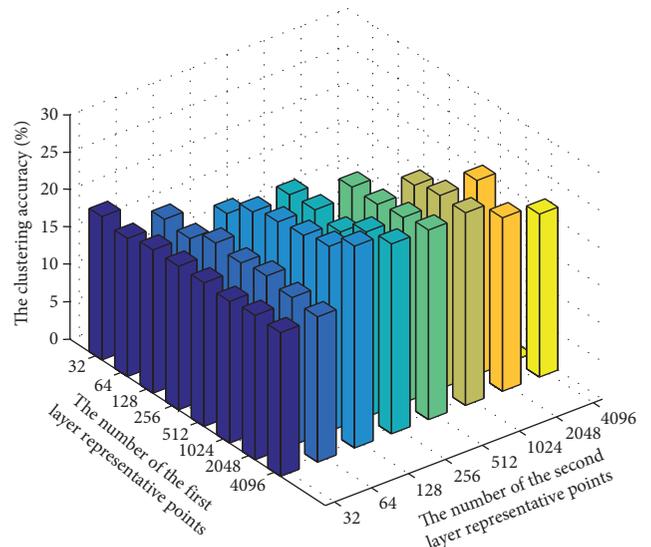


FIGURE 2: Clustering accuracy based on two-layer representative points of Letter.

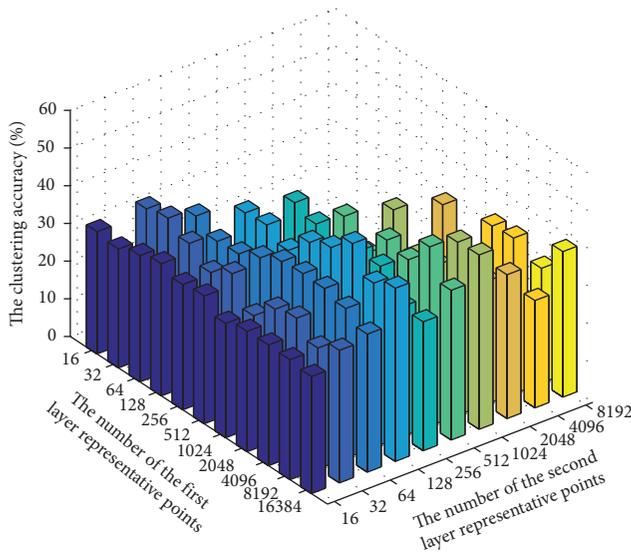


FIGURE 1: Clustering accuracy based on two-layer representative points of Connect-4.

data sets. Table 2 summarizes the clustering time comparison on four data sets. Obviously, SC and LSC-K have high computational cost. KASP has low computational cost on

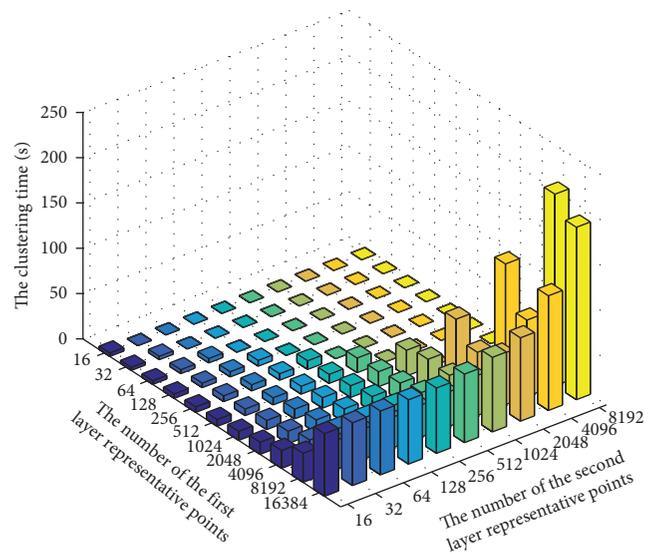


FIGURE 3: Clustering time based on two-layer representative points of Connect-4.

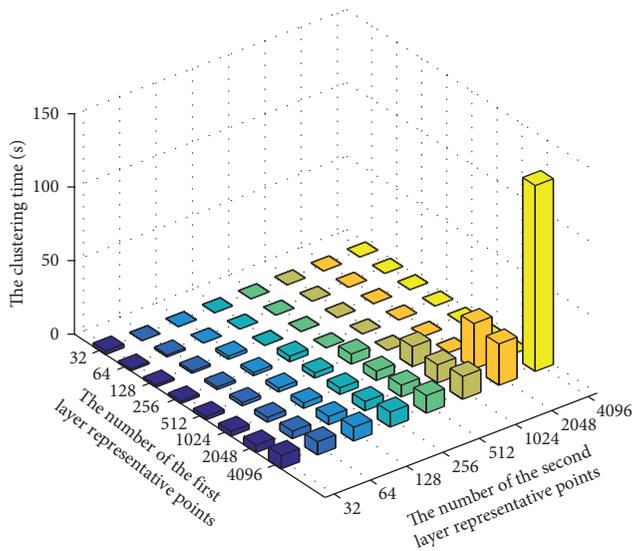


FIGURE 4: Clustering time based on two-layer representative points of Letter.

smaller data sets, but the time cost is very high on large data sets. LSC-R is efficient while the performance is poor. It is easy to see that the proposed RPSC is efficient on all data sets, especially large data sets. Overall, RPSC is the best choice for large data sets among the compared approaches.

Several experiments were conducted on CONNECT-4 and LETTER to demonstrate the influence of the parameter (number of two-layer representative points) for clustering time and performance. Figures 1–4 display that more representative points can increase the clustering accuracy at the beginning, but overly large amounts of representative points are useless. However increasing number of representative points can make the time cost increase.

6. Conclusions

In this paper, we proposed a novel representative point-based spectral clustering approach, named RPSC, based on the two-layer bipartite graph. First, two-layer representative points are generated successively by BKHK. Then, the similarity matrices between adjacent layers are constructed. Different from the conventional kernel-based neighbor assignment strategy, we adopt a parameter-free yet effective neighbor assignment method, which avoids the need to tune the heat-kernel parameter. Furthermore, the final similarity matrix can be easily obtained by multiplying the similarity matrices between adjacent layers. Finally, the coclustering is performed on the final similarity matrix. The coclustering mechanism takes advantage of the cooccurring cluster structure among the last-layer representative points and the original data to strengthen the clustering performance. As a result, the computational complexity can be greatly reduced and the clustering accuracy can be improved. Extensive experiments conducted on 4 large data sets demonstrate the efficiency and effectiveness of the proposed RPSC in terms of computational speed and clustering accuracy. In future work, we will consider designing a more efficient representative point generation algorithm. Furthermore, the clustering ensemble based on the proposed method is an interesting research.

Data Availability

The data used to support the finding of this study are downloaded from the UCI machine learning repository and the LibSVM data sets page.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key Research and Development Project (grant nos. 2017YFC0403600 and 2017YFC0403604), National for Young Scientists of China (no. 61702185), Innovation Scientists and Technicians Troop Construction Projects of Henan Province (2014), Key Scientific and Research Project in University of Henan Province (nos. 15A520021, 15A510003, and 18A520034), Henan Province Science and Technology Research Program (no. 172102210050), Open Research Foundation of Key Laboratory of Sediments in Chinese Ministry of Water Resources (no. 2017001), and Innovation Fund for Ph.D. candidate of North China University of Water Resources and Electric Power (2015).

References

- [1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [2] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1601–1608, Vancouver, Canada, December 2005.
- [3] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.
- [4] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, "Spectral clustering using deep neural networks," 2018, <https://arxiv.org/abs/1801.01587>.
- [5] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.
- [6] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 849–856, Vancouver, Canada, December 2002.
- [7] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pp. 313–318, San Francisco, CA, USA, August 2011.
- [8] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Proceedings of the International Conference on Machine Learning*, pp. 1002–1011, New York, NY, USA, June 2016.
- [9] C. Fowlkes, S. Belongie, F. Fan Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.
- [10] B. Chen, B. Gao, T. Liu, Y. Chen, and W. Ma, "Fast spectral clustering of data using sequential matrix compression," in

- Proceedings of the European Conference on Machine Learning*, pp. 590–597, Berlin, Germany, September 2006.
- [11] M. Li, J. Kwok, and B. Lu, “Making large-scale nystrom approximation possible,” in *Proceedings of the 27th International Conference on Machine Learning*, pp. 631–638, Haifa, Israel, June 2010.
- [12] F. Chung and F. Graham, *Spectral Graph Theory*, American Mathematical Society, Providence, RI, USA, 1997.
- [13] U. von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [14] Y. Wang and L. Wu, “Beyond low-rank representations: orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering,” *Neural Networks*, vol. 103, pp. 1–8, 2018.
- [15] L. Houthuys, R. Langone, and J. A. K. Suykens, “Multi-view kernel spectral clustering,” *Information Fusion*, vol. 44, pp. 46–56, 2018.
- [16] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “Robust spectral ensemble clustering via rank minimization,” *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, pp. 1–25, 2019.
- [17] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, and C. Kwok, “Ultra-scalable spectral clustering and ensemble clustering,” *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [18] J. Yang, L. Sun, and Q. Wu, “Constraint projections for semi-supervised spectral clustering ensemble,” *Concurrency and Computation Practice and Experience*, vol. 31, no. 20, 2019.
- [19] Y. Zhao, Y. Yuan, F. Nie, and Q. Wang, “Spectral clustering based on iterative optimization for large-scale and high-dimensional data,” *Neurocomputing*, vol. 318, pp. 227–235, 2018.
- [20] A. C. Türkmen, “A review of nonnegative matrix factorization methods for clustering,” 2015, <https://arxiv.org/abs/1507.03194>.
- [21] T. Liu, M. Gong, and D. Tao, “Large-cone nonnegative matrix factorization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2129–2142, 2017.
- [22] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. Shen, “Multitask spectral clustering by exploring intertask correlation,” *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1069–1080, 2015.
- [23] L. He, N. Ray, Y. Guan, and H. Zhang, “Fast large-scale spectral clustering via explicit feature mapping,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1058–1071, 2019.
- [24] T. Semertzidis, D. Rafailidis, M. G. Strintzis, and P. Daras, “Large-scale spectral clustering based on pairwise constraints,” *Information Processing & Management*, vol. 51, no. 5, pp. 616–624, 2015.
- [25] H. Shinnou and M. Sasaki, “Spectral clustering for a large data set by reducing the similarity matrix size,” in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pp. 201–204, Marrakesh, Morocco, May 2008.
- [26] D. Yan, L. Huang, and M. I. Jordan, “Fast approximate spectral clustering,” in *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 907–916, Paris, France, June 2009.
- [27] W. Liu, J. He, and S. F. Chang, “Large graph construction for scalable semi-supervised learning,” in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 679–686, Haifa, Israel, June 2010.
- [28] J. Liu, C. Wang, M. Danilevsky, and J. Han, “Large-scale spectral clustering on graphs,” in *Proceedings of the International Joint Conference on Artificial Intelligence AAAI Press*, pp. 1486–1492, Beijing, China, August 2013.
- [29] D. Cai and X. Chen, “Large scale spectral clustering via landmark-based sparse representation,” *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.
- [30] D. Cai, “Compressed spectral regression for efficient non-linear dimensionality reduction,” in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 3359–3365, Buenos Aires, Argentina, July 2015.
- [31] Y. Li, F. Nie, H. Huang, and J. Huang, “Large-scale multi-view spectral clustering via bipartite graph,” in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2750–2756, Austin, TX, USA, January 2015.
- [32] R. Wang, F. Nie, and W. Yu, “Fast spectral clustering with anchor graph for large hyperspectral images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2003–2007, 2017.
- [33] T. Y. Liu, H. Y. Yang, X. Zheng, T. Qin, and W. Y. Ma, “Fast large-scale spectral clustering by sequential shrinkage optimization,” in *Proceedings of the European Conference on Information Retrieval*, pp. 319–330, Rome, Italy, April 2007.
- [34] W. Zhu, F. Nie, and X. Li, “Fast spectral clustering with efficient large graph construction,” in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2492–2496, New Orleans, LA, USA, March 2017.
- [35] F. Nie, X. Wang, M. I. Jordan, and H. Huang, “The cos-trained laplacian rank algorithm for graph-based clustering,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AR, USA, February 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

