INTERNATIONAL JOURNAL · ENGINEERING RESEARCH ·

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 6th Feb 2020. Link :http://www.ijiemr.org/downloads.php?vol=Volume-09&issue=ISSUE-02

Title: ANALYSIS OF GRAPH SPECTROSCOPIC CLUSTERING

Volume 09, Issue 01, Pages: 16-22.

Paper Authors

DEEPTHI KOTHAPETA , SRILATHA KOMAKULA

Chaitanya Deemed to be University

International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

ANALYSIS OF GRAPH SPECTROSCOPIC CLUSTERING

***DEEPTHI KOTHAPETA [1], **SRILATHA KOMAKULA [2]**

* Assistant Professor, Dept. Of Computer Science, Chaitanya Deemed to be University,

** Assistant Professor, Dept. Of Computer Science, Chaitanya Deemed to be University,
deepthivaishu18@gmail.comsrilatha.kom@gmail.com

ABSTRACT

Clustering nodes in a graph is a useful general technique in data mining of large network data sets. In this context, Newman and Girvan [9] recently proposed an objective function for graph clustering called the Q function which allows automatic selection of the number of clusters. In this paper we propose an efficient clustering algorithm for large-scale graph data using spectral methods. The key idea is to repeatedly generate a small number of "super nodes" connected to the regular nodes, in order to compress the original graph into a sparse bipartite graph. By clustering the bipartite graph using spectral methods, we are able to greatly improve efficiency without losing considerable clustering power. Extensive experiments show the effectiveness and efficiency of our approach. 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 co clustering on the final similarity matrix. Co clustering mechanism takes advantage of the co-occurring 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. Unfortunately, these methods usually sacrifice a lot of information of the raw data, resulting in performance degradation. -e traditional spectral clustering needs two independent steps: constructing similarity graph and performing spectral analysis. Both the steps are computational expensive for large-scale data, and their computational complexity is $o(n2)$ and $o(n3)$, respectively. -e ε conventional spectral clustering has three methods to construct the similarity graph which is constructed by pairwise similarities or pairwise distances. -e goal is to model the local neighborhood relationships between the data points. ε 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.

Graph clustering aims to partition the nodes into densely connected subgraphs such that nodes within the same cluster have more connections than those in different clusters. Discovering clusters in graph not only helps to visualize and define hierarchies [Herman *et al.*, 2000], but is also meaningful for many real world problems, such as community detection [Fortunato, 2010; Smyth and White, 2005] and outlier detection [Gupta *et al.*, 2012]. In addition, clustering results can be used as building blocks for many other algorithms to reduce graph and model complexity [Song *et al.*, 2008; Dalvi *et al.*, 2008]. Using and interpreting such methods without some form of summarization becomes difficult as graphs grow in size. However, actually discovering clusters becomes quite challenging as graphs balloon in size, a common phenomenon in today's era of "big data." Thus, there is a pressing need to develop efficient and effective clustering algorithms that can be adapted for large-scale graphs. In this paper, we propose such an algorithm using spectral methods, which have been widely used for effective graph clustering [Shi and Malik, 1997]. Many previous studies have examined accelerating spectral clustering. Most of these [Shinnou and Sasaki, 2008; Yan *et al.*, 2009; Sakai and Imiya, 2009; Chen and Cai, 2011] have been devoted to data

represented in a feature space instead of a graph. Other approaches are designed to achieve efficiency by finding numerical approximations to eigenfunction problems [Fowlkes *et al.*, 2004; Chen *et al.*, 2006; Liu *et al.*, 2007] or adapting standard eigensolvers to distributed architecture [Chen *et al.*, 2011; Miao *et al.*, 2008]. In contrast, we aim to mitigate the computational bottleneck by reducing the size of the graph, while still providing high-quality clustering results, as compared to standard spectral methods. Specifically, we generate meaningful *supernodes* which are connected to the original graph. Correspondingly, we obtain a *bipartite* structure which preserves the links between original graph nodes and the new supernodes. In this representation, we expect these supernodes to behave as cluster indicators that may guide the clustering of nodes in the original graph. Furthermore, the super node clustering and regular node clustering should mutually help induce each other. In this way, the clustering of the original graph can be solved by clustering the bipartite graph. By controlling the number of super nodes and enforcing the sparsity of the generated bipartite graph, we are able to efficiently achieve this goal.

2. RELATED WORK

The general spectral clustering method [Ng *et al.*, 2001; Shi and Malik, 1997] was first shown to work on data represented in feature space. As we are mainly interested in graph data, we need one more step to construct an adjacency matrix which takes $O(n2p)$ time where n and p represent number of data points and features respectively. Calculating the eigen decomposition of the corresponding Laplacian matrix is the real computational bottleneck, requiring $O(n3)$ time in the worst case. Therefore, applying spectral clustering for largescale data becomes impossible for many applications. In recent years, many works have been devoted to accelerating the spectral clustering algorithm.

Among them, [Fowlkes *et al.*, 2004] adopts the classical Nystr¨om method, which was originally proposed to find numerical approximations to eigenfunction problems. It chooses samples randomly to obtain small-size eigenvectors and then extrapolates these solutions. [Shinnou and Sasaki, 2008] reduces the original data set to a relatively small size before running spectral clustering. Similar to this idea, in [Yan *et al.*, 2009], all data points are collapsed into centroids through *k*-means or random projection trees so that eigen-decomposition only needs to be applied on the centroids. [Sakai and Imiya, 2009] uses random projection in order to reduce data dimensionality. Random sampling

has also been applied to reduce the size of data points within the eigen-decomposition step. [Chen *et al.*, 2006; Liu *et al.*, 2007] introduce early stop strategies to speed up eigen-decomposition based on the observation that wellseparated data points will converge to the final embedding more quickly. In [Chen and Cai, 2011], landmark points are first selected among all the data points to serve as a codebook. After encoding all data points based on this codebook, acceleration can be achieved using the new representation. The authors in [Khoa and Chawla, 2012] work on resistance distance embedding, which employs a similar idea to spectral clustering and exhibits comparable clustering capability.

To tackle the problem, a novel and efficient representative point-based spectral clustering method is proposed to deal with large-scale data sets. -ree 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.

3. LARGE-SCALE SPECTRAL CLUSTERING ON GRAPHS

Now we introduce our *Efficient Spectral Clustering on Graphs (ESCG)* for large-scale graph data. The basic idea of our approach is designing an efficient way to coarsen the graph by generating *supernodes* linked to the nodes in the original graph. A bipartite graph between nodes in *G* and generated supernodes is then constructed to replace *G*, so that the original high-dimensional EVD can be avoided.

3.1 Generation of Supernodes

Given the initial graph *G* of *n* nodes, we want to generate a set of *d* supernodes to coarsen the graph under the condition that $d \ll n$. Inspired by the intuition behind *simultaneous* or *co-clustering* [Dhillon, 2001], which says that clustering results of two related object types can be mutually enhanced, we expect that a partition of supernodes can induce a partition of the observed nodes, while a partition of the observed nodes can imply a partition of supernodes. Therefore, we first develop a simple and efficient algorithm to establish an initial clustering on graph *G*. Then we generate supernodes based on this initial clustering.

Our proposed approach works as follows: given the graph *G*, we randomly pick *d* seeds in the graph and compute shortest paths from these seeds to the rest of the nodes. We then partition all the nodes into *d* disjoint subsets represented by the seeds: each node chooses the representative seed with the shortest distance.

To solve the shortest path problem, we first transform the edge weight demonstrating the similarity into distance:

$$M_{ij} = -\log \frac{W_{ij}}{\max W} + \varepsilon$$

where ε is a very small number that functions as the additional decay along the path. We incorporate the decay in order to prevent the possible distance between two nodes from being 0. The logarithmic transformation is adopted because the summation of Mij ’s along the paths in the graph can be viewed as the multiplication of the edge weights, which makes sense for estimating the distance for any pair of nodes. After this step, the range of the distance value on each edge should be within [ε,+∞). Dijkstra’s algorithm is then adopted to compute the shortest paths from seeds to the rest of the graph, which takes O(md + nd log n) time. After running Dijkstra’s algorithm, we are able to partition all nodes in *G* into *d* disjoint subsets by comparing their shortest paths to the seeds. We assign each node to the partition with the closest seed. Note that this step can be implemented in parallel.

Spectral Clustering on Reduced Graphs

Through the transformation to the bipartite graph, we significantly reduce the size of the full edge weight matrix of G from $n \times n$ to $d \times n$. In this section, we introduce how to convert the EVD of the graph Laplacian mentioned in the previous section into a singular value decomposition (SVD) problem, such that the overall time complexity is $O(md + nd \log n + nd2)$, which is a significant reduction from $O(n3)$ since $d \ll n$.

To begin, we give the adjacency matrix of the bipartite graph described above:

$$W' = \begin{bmatrix} 0 & \hat{W}^T \\ \hat{W} & 0 \end{bmatrix}.$$

Here, we use the “prime” notation to denote the adjacency matrix, which is a square matrix of size $(n + d) \times (n + d)$.

Hence we also have representations for LL' and D'in this bipartite model:

$$L' = \begin{bmatrix} D_1 & -\hat{W}^T \\ -\hat{W} & D_2 \end{bmatrix}, \text{and} \quad D' = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

where $D1$ and $D2$ are two diagonal matrices whose entries are column and row sums of ˆW , respectively.

3.2 Regeneration of Supernodes

In the aforementioned approach, supernodes are connected to regular nodes according to the shortest paths to randomly selected seeds. The supernodes thus behave like cluster indicators responsible for propagating knowledge of the original graph. However, once we get the clustering result using spectral techniques discussed above, we may use this knowledge to form non-random supernodes and improve the final results. We therefore propose an iterative way to regenerate the supernodes based on the current clustering results, aiming to repeatedly improve the clustering. In particular, it is natural to require each supernode to link to a set of densely connected nodes, which themselves form a better local cluster than the random sampling method discussed in Sec. 4.1. Also the process of discovering such local clusters must be efficient. Inspired by the fact that the column vectors of the embedding matrix U can be used to indicate partitions of nodes in the graph [Shi and Malik, 1997], supernodes can be guided to connect to nodes which form local clusters that are inferred from the element values in the column vectors of U.

4. COCLUSTERING ON SIMILARITY MATRIX

4.1 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. -e kernel-based neighbor assignment strategy usually is sued in conventional methods, but it always brings extra parameters [13]. A parameter-free method is adopted in this paper [35]. Let $U \in Rm\times d$ denote the generated representative points, and $U\langle i\rangle$ is the set of k-nearest representative points for the i-th sample.

As the same as document data, duality exists between the raw data points and the representative points. -e 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.

4.2. 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:

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

$$D = \begin{bmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix},$$

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

CONCLUSIONS

In this paper, we proposed a novel representative point-based spectral clustering approach, named RPSC, based on the twolayer bipartite graph. First, two-layer representative points are generated successively by BKHK. -en, the similarity matrices between adjacent layers are constructed. Although graph compression naturally induces inaccuracy, empirical studies demonstrate that our method can considerably decrease the necessary runtime while posting a tolerably small loss in accuracy. We propose a method to reduce graph size, based on effectively compressing the graph information into a smaller number of "supernodes". Clustering supernodes with spectral methods is less expensive, and the clustering results can also be propagated back to the original graph with low cost.

REFERENCES

[1] C. Alpert and S. Yao, Spectral partitioning: the more eigenvectors the better. In *Proceedings of 32nd ACM/IEEE Design Automation Conference*, 1995, pp. 195-200.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Vorst, eds., Templates for the Solution of Algebraic Eigen-value Problems: A Practical Guide, SIAM, Philadel- phia, 2000.

[3] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. *9th International Conference on Arti¯cial Intelligence and Statistics*, 2002.

[4] F. Chung. Spectral graph theory. Number 92 in *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[5] C. Elkan. Using the triangle inequality to accelerate k-Means. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 147-153.

[6] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23 (1973), pp. 298-305.

[7] K. Hall. An r-dimensional quadratic placement algo- rithm. *Management Science*, 11(3)(1970), pp. 219-229.

[8] G. Miller. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3 (1990), pp. 235-312.

[9] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 026113 (2004).

[10] M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 066133 (2004).

AUTHOR 1:

NAME: DEEPTHI KOTHAPETA
QUALIFICATIONS: M.Sc(CS).,M.Tech
DESIGNATION: Assistant Professor
DEPARTMENT: Computer Science
Chaitanya Deemed to be University

AUTHOR 2:

NAME: SRILATHA KOMAKULA
QUALIFICATIONS: MCA.,M.Tech.
DESIGNATION: Assistant Professor
DEPARTMENT: Computer Science
Chaitanya Deemed to be University