Transdisciplinary Engineering: Crossing Boundaries
M. Borsato et al. (Eds.)

doi:10.3233/978-1-61499-703-0-573

A Distance-Based Spectral Clustering Approach with Applications to Network Community Detection

Gang SHEN and Dongmei YE
School of Software Engineering, Huazhong University of Science and Technology
Wuhan, China 430074

Abstract. Object clustering is a fundamental task in many data analysis and pattern understanding applications by providing insights into detecting the underlying structures of a large collection of samples. In this paper, we present our work on a novel spectral clustering algorithm that partitions a collection of objects using the spectrum of adistance matrix. If the nodes in a metric space can be associated with a well defined distance, the distance matrix is almost negative definite, implying that the eigenvector for the smallest eigenvalues of this matrix can be used as an approximation of the solution to a quadratic form partition problem. It is proved that this smallest eigenvalue is equivalent to the second largest singular value. Therefore Lanczos iterative algorithm can be applied to findingthe eigenvalues efficiently. We adapted this algorithm to the distributed network community detection problem using a decentralized multi-agent framework, and tested the effectiveness of the proposed approach with simulations.

Keywords. Spectral clustering, network community detection, multi-agent systems, almost negative definite matrices, Lanczos algorithm

Introduction

The rapid penetration of the mobile Internet has facilitated the growing popularity of numerous social networks and their respective applications, enabling social networks to play increasingly important roles in disseminating information and aggregating public opinions. In practice, many social networks can be effectively modeled as multi-agent systems, in which each agent only interacts locally with a small number of other agents. Within many social networks, the direct connections among the network users are established by personal ties in the first place, thus forming some possibly overlapped circles of friends in different sizes. In general, community structures exist in various networks, and the communities may evolve with memberships changing over time. Detecting community structures is a challenging problem in many scientific and engineering fields and various algorithma have been proposed to solve tis problem in different contexts[1,2]. Because there is no centralized coordinating entity existing in social networks, finding individuals with shared interest or similar pattern is not a simple task. In this paper, we intend to investigate a distributed way to allow the users to detect the networks communities consisted of similar users only using the local communications.

In the fields of data mining and machine learning, clustering is a critical step to process the data samples for the purpose of dimensional reduction or pattern detection. Though there is no universally accepted precise definition for the term clustering, clustering can be roughly treated as grouping the similar objects while separating the dissimilar ones. Many clustering techniques have been developed over years, including k-means, hierarchical clustering, density-based clustering and graph-based clustering, just to name a few. With the help of the singular value decomposition (SVD) and graph cut theory, spectral clustering makes use of the distance/similarity matrix of the dataset by examining its eigenvalues and eigenvectors. In different settings, large eigenvalues or the second smallest eigenvalue provide insights into finding cluster structures in the data[3,4]. In [3], the eigenvectors corresponding to the top eigenvalues are used to approximate the data points in a lower dimensional space and k-means algorithm can be applied on top of this result. In [4], the eigenvector for the second smallest eigenvalue of the symmetric normalized Laplacian matrix is used to partition the data into two subsets and hierarchical partitioning can be carried out within each subset. Either way, the number of clusters, k, or the quality of clusters must be predefined to make the process feasible. For example, in [5] the authors proposed an (α, ϵ) measure for the clustering quality, and [6] presented a self-tuning optimization framework to find the best-fit number of clusters. However, making assumption on the parameters is a subtle issue when the data distribution may vary greatly from case to case. Another interesting development related to spectral clustering is the decentralized versions of the spectral analysis algorithms by computing the eigen-decomposition of the weighted adjacency matrix with power iteration, QR decomposition or Lanczos algorithm in a distributed fashion[7, 9-11].

The work in this paper is motivated by the following facts: first, because of the large size of the social network and the need of privacy protection, it is unrealistic to rely on a single node of authority in the network to detect the similarity based communities and inform the rest of the network of the result[7]; second, in multi-agent systems, many collective objectives can be achieved by local interactions and simple evolutionary dynamics; finally, there are extensive applications of consensus algorithms in networked systems[8,9]. The contributions of this paper are: we propose a novel approach that makes use of the eigenvectors of the distance matrix that are corresponding to the negative eigenvalues to bisect the set of nodes recursively by introducing two partitioning quality indices for single eigenvector and cross eigenvectors, without the need to estimate explicitly the number of clusters; in the mean time, we introduce a distributed algorithm that is able to perform the spectral analysis for the distance matrix (not identical to adjacency matrix) overlaying the communication network, without exchanging the distance information between nodes.

The rest of the paper is organized as follows. In Section 1,we formulate the problem and present the definitions and algorithms used for distance matrix based spectral clustering; then in Section 2, we discuss the design of the distributed version of the Lanzcos algorithm that prepares the necessary data for each node to find the identical clustering outcome simultaneously. The experiments for some synthesized data and benchmark data are provided and analyzed in Section 3, showing the effectiveness the proposed approach. And finally in Section 4 we conclude this paper by remarking the proposed future research.

1. Problem formulation and the proposed clustering approach

In a metric space, pairwise distance is defined for points in that space, satisfying the following properties: symmetric, nonnegative and the triangle inequality. It is reasonable to assume that any user in a social network is able to find the distance between any other user and itself, provided that the users not directly interacting may observe the states and behaviors of others. We do not need the specific description of the space, as long as the distance metrics can be measured by the users on their own. Given a set of n users $\{x_1, x_2, ..., x_n\}$, and their distance measures (represented by the distance matrix A, where $a_{ij} = d(x_i, x_j)$), the clustering objective is to partition all the users into a number of disjoint subsets (assuming this number is unknown a priori), such that the members in the same subset are close in distances, and the members belonging to different subsets are not as close. One can find many different versions of the clustering approaches, in this paper, we adopt the partition cost function

$$\begin{array}{ll} \min_y & y^T A y \\ \text{subject to} & y_i \in \{-1, 1\}, 1 \le i \le n \end{array} \tag{1}$$

to find the two subsets of users, with -1 and 1 being the respective membership values. Since the above optimization problem is hard to solve, we take the similar spectral heuristics as in the normalized cut method [4]. Because the smallest negative eigenvalue of the matrix A, λ_n, is the minimum of A's Rayleigh quotient problem

$$\min_z \frac{z^T A z}{z^T z} \tag{2}$$

we will use u_n, the normalized eigenvector to find the approximate solution to (1). However, we will not take the hierarchical partitioning to find the clusters as proposed in [4], to avoid the repeated eigen-decompositions for reduced distance matrices. Instead, we wish we could find the final cluster structures with the eigen-pairs of the original A.

Lemma 1. Given a well-defined distance matrix $A_{n\times n}$, there exists a set of points of points in $n-1$ dimensional Euclidean space, $\{p_1, p_2, ..., p_n\} \in R^{n-1}$, such that. $d(p_i, p_j) = a_{ij}$.

Definition 1[12]. A real symmetric matrix $M$with zero diagonal entries is said to be almost negative definite if $c^T M c < 0$ for all vectors $c \in R^n$ satisfying $c^T \mathbf{1} = 0$, where $\mathbf{1} = [1, 1, ..., 1]^T$.

It is proved in [12] that for a set of points $\{x_1, x_2, ..., x_n\} \in R^d$, the matrix $D = \{d^2(x_i, x_j)\}$ ($d(x_i, x_j)$is the Euclidean distance between x_i and x_j) is almost negative definte.

Lemma 2. If $A = \sqrt{a_{ij}}$ is a well-defined distance matrix, $A = \sqrt{a_{ij}}$ is also a well-defined distance matrix.

Therefore we have the following property for the distance matrices.

Theorem 1. The matrix $A_{n\times n} = \{a_{ij}\}$ is almost negative definite.

Theorem 2. A nonsingular $A_{n\times n} = \{a_{ij}\}$ has $n-1$ negative eigenvalues.

Because A has only 0's on its main diagonal, $\sum_{i=1}^{n} \lambda_i = 0$, thus we have the following conclusion.

Corollary 1. Let the distance matrix A have eigenvalues$\lambda_1 > \lambda_2 > \ > \lambda_n$, $|\lambda_n|$is the second largest singular value of A.

Denote a column of A as $a_i = [a_{i1}, a_{i2}, ..., a_{in}]^T$, $a_i \in R^n$ can be considered as a point in a high dimensional Euclidean space. Though in this paper, we will not apply the algorithms such as MDS to find a low dimensional space to embed these points, it is rather straightforward to conclude that the eigenvectors corresponding to the larger singular values of A constitute a subspace that approximates the coordinates of the related points. It is worth mentioning that the leading singular value plays a less important role than the singular values next to it in partitioning data points into clusters because the spread of its eigenvector components is relatively small due to the fact that all its components have the same sign (see Figure 1). In this sense, $\lambda_n$is the most significant eigenvector in partitioning data points into clusters.

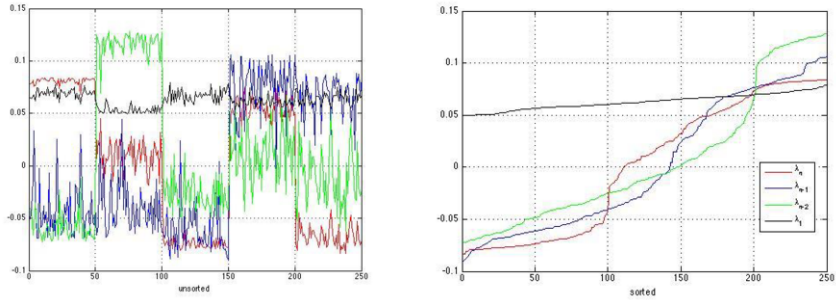

Figure 1.Unsorted (left) and sorted (right) eigenvectors corresponding to the top 4 singular values, black is for the leading one.

Corollary 2. Let a_i be the i[th] column of the distance matrix A, when the eigenvector components of the leading eigenvalue are identical, i.e., $u_{1i} = u_{1j}, i \neq j, 1 \leq i, j \leq n$, then we have the bounds $(u_{ni} - u_{nj})^2 \lambda_n^2 \leq ||a_i - a_j||^2 \leq \max_k (u_{ki} - u_{kj})^2 \lambda_1^2$.

According to the above results, we attempt to partition the set into the two based on the maximal gap between two subsets of eigenvector components. The first step is to sort the eigenvector entries of u_n, then we examine the differences between a pair of adjacent entries in the sorted eigenvector for the maximum, instead of using whether $u_{ni} > 0$ as the criterion as in the standard normalized cut[4] to avoid separating a proper cluster in the middle. However, if there are outliers located far away from the mass of data points, the maximal gap could lie between these outliers and the rest of the data, resulting in trivial clustering. To prevent singling out only the outliers, we expect that the best cut should generate practically balanced groups of data by introducing a cut quality index for eigenvector u_k.

Definition 2 (intra-vector cut quality). The cut quality for a pair of components i and j neighboring in values for a sorted of eigenvectoru_k's components is defined a

$$q_k^{i,j} = \begin{cases} \frac{|u_{ki} - ukj| \cdot |\{u_{kl} : u_{kl} \le u_{ki}\}| \cdot |\{u_{km} : u_{km} \ge u_{kj}\}|}{r_k n^2} & \text{if } \nexists l, \text{such that } u_{ki} \le u_{kl} \le u_{lj} \\ 0 & \text{otherwise} \end{cases}$$

where $|\{\cdot\}|$ represents the cardinality of a set, and $r_k = \max_i u_{ki} - \min_i u_{ki}$is the range of all components in u_k. The best cut for u_k is denoted $q_k^* = \max_{1 \le i,j \le n} q_k^{i,j} r_k$.

In Figure 2 are the best partition results of the few eigenvectors for the smallest eigenvalues of the distance matrix, obtained with the help of the cut quality as defined in Definition 2.

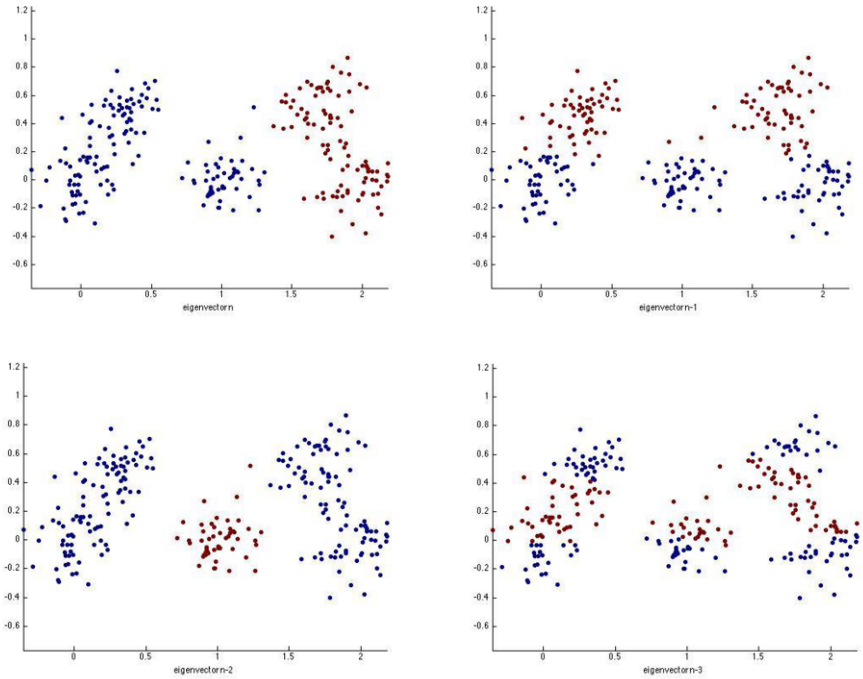

Figure 2.The best partition results based on eigenvectors for 5 Gaussian clusters.

The intersections of these partitions generated by the properly selected eigenvectors are plausible candidates for the clustering results. Since u_n achieves the optimum of problem (2), it may naturally separate data points into multiple different clusters to a greater degree of robustness than the other eigenvectors. It is attracting to take each of the eigenvectors associated with large negative eigenvalues to infer more than one cut of the dataset. As faced by other hierarchical cutting techniques, continuing to refine the cuts has to stop when certain condition is met, but this condition depends on the way to define a cluster, which is also a challenging issue in nature. In order to not explicitly specify the measurement of the cluster quality with the distance/similarity information of all points, we desire to rely only on the eigenvectors per se. Consequently, we propose a quality index to compare cuts between different eigenvectors, combining Corollary 2 and Definition 2, i.e., $q_k^{i,j} \cdot |\lambda_k|$. By doing so, we may find the successively cuts of the data with respect to a given eigenvector u_k, as long as these cuts are better than $q_{k-1}^* |\lambda_{k-1}|$, the best quality rendered by the eigenvector next to it.

Since we limit ourselves to performing data clustering exclusively dependent on the eigen-decomposition of the distance matrix, to find the exact number of clusters remains a critical challenge. We will not estimate this number in the proposed approach, instead we assume that the minimal size of a cluster (μ) and the maxmal number of clusters (M) are available a priori. Specifically we base the procedure on the following observations: first, when the eigenvalues become insignificant in magnitude, they present a pattern of disorder and will cut almost all found clusters into two, making the number of intersections nearly doubled (to be orthogonal to the eigenvectors of eigenvalues with large absolute values, see Figure 2), and lead to the poor cut quality by splitting every cluster into two subsets; second, when the magnitudes of some eigenvalues are not sufficiently small and may be able to result in acceptable partitions, they will not add new value to the whole procedure since the partitions made by the earlier cuts with respect to eigenvectors corresponding to large magnitude eigenvalues (given the sensitivity of the eigenvectors for eigenvalues close to zero, they may bring in additional noise by creating new intersections with a few data points other than only repeat what has already been found). Therefore, if we make practical assumptions on the smallest size of a cluster, as well as the maximal possible number of clusters, we then can force the clustering routine to stop if either the number of possible clusters would have been doubled and exceed the given bound, or the new cut would not give new clusters. This is summarized into the proposed algorithm as listed in Table 1.

Table 1. Distance matrix based spectral clustering algorithm.

Step 1	set μ and M initialize the cluster set as the containing only singleton, $C = \{x_1, x_2, ..., x_n\}$
Step 2	for i = n down to 2 sort u_i by the entries in descent order for j = 1 to n/2 + 1 calculate $q^i_{j,j+1}$ store the current maximum to q^*_i end end
Step 3	for i=n down to 3 for j = 1 to n-1 find j such that $q_i^{j,j+1} * \lambda_i > q^*_{i-1}\lambda_{k-1}$ use j found above to cut the dataset into two disjoint subsets find the intersections with cardinality $> \mu$ of the subsets just found and C if no new intersection is found continue end if the number of intersections $> M$ go to step 4 end set the intersections as members of C end end
Step 4	return C

In practice, only a few eigenvectors are needed in clustering because in general the eigenvalues dwindle fast in magnitude.

2. Distributed spectral algorithm

In this section, we discuss the design and implementation of a distributed approach that allows the users in a connected social network to finish the spectral analysis of the distance matrix locally and use the eigen-decomposition results to find all the clusters independently, without collecting distance information from other users. This objective is achieved by applying average consensus algorithm to exchange the necessary data to construct the tri-diagonal matrix with the help of Lanczos algorithm[13]. It is known that by introducing a good weight matrix, a multi-agent system may reach distributed consensus or average consensus. The problem setup in this paper assumes that the communication network topology may not be coincidental to the distances determined by the other user features. We also suppose that any user is capable of finding the distances to others, but will not share the data. In order to implement the average consensus algorithm, we require each user to select $K > 0$ neighbors using the existing topology until a connected network thus built is confirmed. Once the symmetric double stochastic weight matrix gets set, the information exchange process is characterized by the following dynamics: $x_i(k+1) = \frac{1}{K+1}[x_i(k) + \sum_{j \in N_i} x_j(k)]$. It follows that $\lim_{k\to\infty} x_i(k) = \frac{\sum_{j=1}^{n} x_i(0)}{n}$.

In the iterations of Lanczos algorithm, the only information that needs the coordination of different users is the vector AQ. Because each user i has only distances measured between itself and other users, the user is able to update just the component $a_i^T Q$. Since the weight matrix is chosen to achieve average consensus, it is expected that the users will converge to the vector $\frac{\sum_{i=1}^{n} X_i(0)}{n}$, where $X_i(0)$ represents the initial vector at the user i. If a user only sets $X_{ii}(0) = a_i^T X(0)$, and all other $X_{ij} = 0$, then when the consensus is reached, each user has $X_i = AX(0)$. This process is listed in Table 2, where the inputs of maximal steps K and the error bound ϵ are required.

Table 2. Distributed average consensus algorithm for user i.

Step 1	user i selects K neighbors $\mathcal{N}(i)$, set W(i,j)=1/(K+1), if j=i or $j \in \mathcal{N}(i)$ initialize vector $X_{i,i}(0) = x(0), X_{i,j} = 0$ send $X_i(0)$ to neighbors in $\mathcal{N}(i)$ set $k = 0$
Step 2	update $X_i(k+1) = \sum_{j=i or\ j \in \mathcal{N}(i)} w(i,j) * x_j(k)$
Step 3	if $\|\|X_i(k+1) - X_i(k)\|\|^2 < \epsilon$ go to Step 4 if $k > M$ go to step 4 set $k = k+1$ go to Step 2
Step 4	return nX_i

Therefore, by applying this simple average consensus process, the Lanczos iterations can find a tri-diagonal matrix T that is similar to A (see Table 3). The eigen-decomposition for T is relatively cost-effective and can be performed by an individual user. After the eigenvalues/eigenvectors are available to each user's, the spectral clustering proposed in Section 1 will generate the same results and the users then can decide the clusters to which they belong. Lanczos algorithm may terminate at iterations fewer than n when we accept approximate solutions to eigenvectors for the most significant eigenvalues.

Table 3. Decentralized Lanczos algorithmfor AX at user i.

Step 1	user I has its distances to other users measured, and knows the size n initialize $R_0 = \mathbf{1}/\sqrt{n}, \beta_0 = 1, Q_0 = \mathbf{0}, k = 0$
Step 2	$Q_{k+1} = R_k/\beta_k$ set $k = k + 1$ apply average consensus to get $V = AQ_k$ $\alpha_k = Q_k^T V$ $r_k = V - \alpha_k Q_k - \beta_{k-1} Q_{k-1}$ $\beta_k = R_k^T R_k$
Step 3	if $\beta_k \neq 0$ and $k < n$ go to Step 2
Step 4	return $T = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix}$

3. Experiments and analysis

In this section, we look at the performance of the proposed approach by applying it to different datasets. Though the porposal was first designed to solve clustering problems for mixed Gaussian data (balls in space, as shown in Figure 2), it also works for other data as long as the clusters are separated well by between-cluster distance (see Figure 3) in a properly designed distance measure. It is noted that the Euclidean distance used in Figure 3 is not a proper choice for these connected structures, nonetheless the proposal is able to find the correct clusters since the eigenvector u_n has sufficient gaps between different groups of entries.

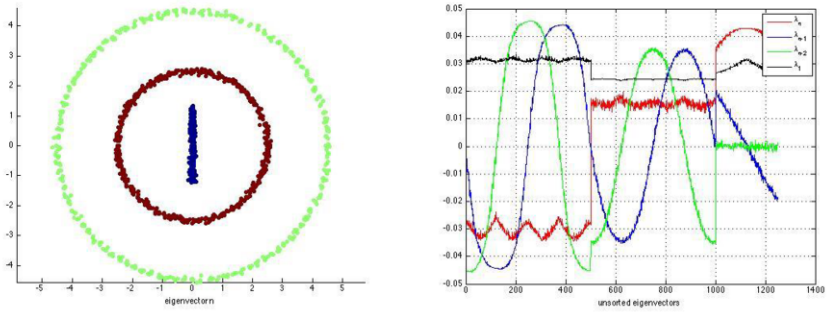

Figure 3.Clustering result using eigenvector u_n (left) and the distribution of the top 4 significant eigenvectors (right).

An interesting question is the choice of distance measures in different applications. We only considered the most straightforward way to define distance in the tests in this paper and believe more advanced distance functions will reach more accurate results. One example is the typical classification benchmark, the Iris dataset. In the test, we used the Euclidean distance of the five normalized attributes. Clearly, the proposed

approach found the correct number of clusters, while Iris-versicolor and Iris-verginica got mixed results due to the closeness in distance (Table 4), and Iris-sentosa stood out of the rest.

Table 4 Confusion matrix for Iris classes.

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	98%	0%	2%
Iris-versicolor	0%	88%	12%
Iris-virginica	0%	28%	72%

The distance used in the Thackeray Karate club dataset is the pairwise shortest path with each link assigned a unit distance. To test the distributed algorithm, we applied the average consensus algorithm to pass the messages between nodes. Four clusters were found in this case, as shown in Figure 4.

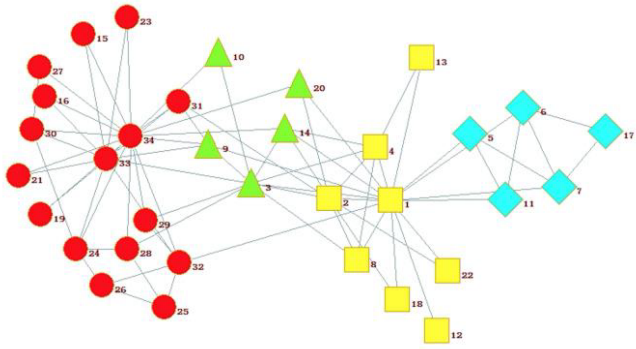

Figure 4.Clustering result of Karate club.

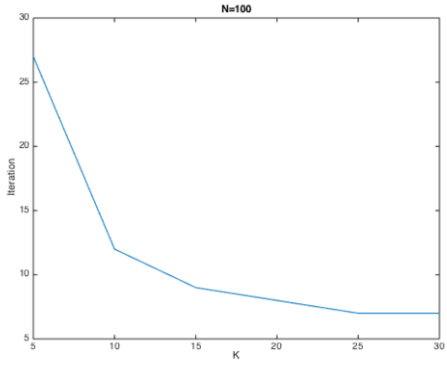

Figure 5.Convergence speed of average consensus in number of iterations, given 100 nodes.

As shown in Figure 5, the number of iterations needed to make consensus in a netwrok of 100 nodes is related to the connectivity. If each one is connected to more than 10% of all nodes, the average consensus can be reached in roughly 10 iterations.

4. Concluding remarks

The users of social networks may derive similarities based on the their attributes other than the existing connectivity, therefore, evolve into new communities with similar

members on top of the communication topology is an important phenomenon observed in many of today's social network applications. We adapt the spectral clustering framework to a decentralized scheme. In general, a user may not be willing to share sensitive information with others. In order to let individual users detect the global cluster structures, we apply average consensus algorithm and Lanczos iterations to allow the users to exchange only the necessary messages.In this paper, we first proposed a distance matrix based clustering approach that makes use of its eigenvectors corresponding to the significant negative eigenvalues. We also presented ourinvestigation on evaluate the partition quality within an eigenvector (balancing the gap sensitivity and cluster size) and between eigenvectors (considering both the gap and eigenvalue). We developed the algorithm that can have multiple cuts on a single eigenvector and use the intersections of different cuts to form clusters. Then we discussed the conditions to make the partition process to terminate, using the reasonable assumptions on the minimal size of a cluster and the maximal possible number of clusters. In the simulations, we tested both synthesized and benchmark datasets, showing that the proposed approach worked effectively in both cases. However, in this paper, we suppose that the users have the global observation on others, which may not be true for all situations. Second, the partition quality is designed to separate clusters without overlapping. Third, while it is acceptable to run the distributed calculation in a synchronous way, it is much helpful to let the users detect the communities asynchronously because realistically the users may join or leave the social networks at ant time. These will be attractive subjects for our future research.

References

[1] Fortunato S. Community detection in graphs, *Physics reports*, 2010, 486(3): 75-174.

[2] J. Leskovec, K.J. Lang and M. Mahoney, Empirical comparison of algorithms for network community detection, *Proceedings of the 19th international conference on World wide web*, ACM, 2010, pp. 631-640.

[3] A.Y. Ng, M.I. Jordan and Y. Weiss, On spectral clustering: Analysis and an algorithm, *Advances in Neural Information Processing Systems*, 2002, No. 2, pp. 849-856.

[4] J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, No. 22(8), pp. 888-905.

[5] R. Kannan and S. Vempala, A. Vetta, On Clusterings : Good. Bad and Spectral, *Journal of the ACM*, Vol 51, 2004, No. 3, pp. 497–515.

[6] L. Zelnik-Manor and P. Perona, Self-tuning spectral clustering, *Advances in Neural Information Processing Systems*, 2004, pp. 1601-1608.

[7] D. Kempe and F. McSherry, A decentralized algorithm for spectral analysis, *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, ACM, 2004, pp. 561-568.

[8] P. Frasca, R. Carli, F. Fagnani and S. Zampieri, Average consensus by gossip algorithms with quantized communication, *47th IEEE Conference on Decision and Control*, CDC 2008, pp. 4831-4836.

[9] F. Penna and S. Stanczak, Decentralized eigenvalue algorithms for distributed signal detection in wireless networks, *IEEE Transactions on Signal Processing*, 2015, 63(2), pp. 427-440.

[10] M. Jelasity, G. Canright and K. Engø-Monsen, Asynchronous distributed power iteration with gossip-based normalization, *Euro-Par 2007 Parallel Processing*, LNCS 4641, Springer Berlin Heidelberg, 2007, pp. 514-525.

[11] N.D. Thang and Y.K. Lee, S. Lee, Deflation-based power iteration clustering, *Applied Intelligence*, 2013, 39(2), pp. 367-385.

[12] C.A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximation*, 1986, 2(1), pp. 11-22.

[13] M. Panju, Iterative methods for computing eigenvalues and eigenvectors, *arXiv preprint arXiv*:1105.1185, 2011.