Clustering protein-protein interaction data with spectral clustering and fuzzy random walk

E Krisna, A Bustamam, and K A Sugeng

Department of Mathematics, Universitas Indonesia, Depok, Indonesia

Email: edo.krisna21@sci.ui.ac.id, alhadi@sci.ui.id, kiki@sci.ui.ac.id

Abstract. Spectral Clustering is a graph clustering algorithm that makes use of eigenvector obtained from a matrix describing pairwise similarity between data points. It provides a dimensionality reduction for clustering in lower dimensions. One example of spectral clustering application is the clustering of protein-protein interaction (PPI) network. PPI networks are usually represented as a graph network with proteins and interactions as vertices and edges respectively. However, this spectral clustering only produces a hard clustering of proteins, whereas there may be some relationship between each protein clusters, and possibly multiple functionality for each proteins that has not been detected before. Fuzzy Random Walk is a fuzzy clustering method based on transition probability from a random walk on a dataset. In this paper, we combine both Spectral Clustering and Fuzzy Random Walk to cluster PPI network of protein TP53, a protein that plays an important role in managing cell cycle, especially in tumor cell suppression. Using PPI dataset of TP53 obtained from the STRING database, we found the combined algorithm is proven to produce both robust and fuzzy clusters with each cluster explains one of TP53 protein’s functionality related to the tumor cell.

Keywords: Spectral Clustering, Protein-protein interaction (PPI) network, Fuzzy Clustering, Fuzzy Random Walk, TP53 Protein, Tumor cell.

1. Introduction

The increasing complexity and heterogeneity of a data make it difficult to interpret and extract possibly hidden information within the data. This problem has led the development of many dimensionality reduction and data clustering techniques over past decades. Clustering is a process of revealing the underlying structure of a dataset by grouping the data elements into a groups called clusters [1]. Clustering is usually done based on some similarity measure between data elements, and without any information on how the grouping should be done (unsupervised learning) [2].

Graph is a structure formed by a set of vertices(also called nodes) and a set of edges that connect the pairs of vertices. Graph clustering is a task of grouping the vertices of the graph into some clusters, by considering the edge structure of the graph, in such a way that there should be many edges within each cluster and relatively few between the clusters [3]. One of the popular graph clustering techniques are spectral clustering. Spectral clustering provides a dimensionality reduction by using the eigenvectors of a similarity matrix of the graph [4]. Spectral clustering is a global clustering method, as it considers all of the graph vertices in its algorithm.
Protein-protein interaction (PPI) is a physical contact of high specificity between two or more proteins as a result of biochemical activity steered by electrostatic force including hydrophobic effect [5]. PPI plays a significant role in almost all biological process [5]. Clustering of PPI network is an effective approach in identifying protein complexes and functional module, which is important in investigating a gene role in a disease and also drug target [6]. Broadly speaking, this approach could also give a new understanding about protein function and cell machinery. Spectral clustering of PPI data gives not only a robust clusters, but also possibly some meaningful interpretation in the low-dimensional embedding [7]. However, spectral clustering only produces a ‘hard’ clusters, whereas there may be some relationship between each protein clusters, and possibly multiple functionality for each protein that has not been detected.

In this paper, we propose a fuzzy version of spectral clustering by combining spectral clustering with fuzzy random walk [8,9] and apply them to PPI network of protein TP53, an important protein for tumor suppression. We set a ‘seed’ or cluster center for each clusters and perform a random walk for each data to determine its membership value. Using PPI datasets of TP53 obtained from the STRING database, we found the combined algorithm is proven to produce both robust and fuzzy clusters with each fuzzy cluster explains each one of TP53 protein’s functionality related to the tumor cell.

2. Methods
Our proposed algorithm combines the concept of spectral clustering and fuzzy random walk along with its specific modification for PPI dataset. The input of the algorithm is a graph \( G = (V,E) \) with \( n \) nodes \( 1,2,3,...,n \). There are some noticeable changes between our method and the original fuzzy random walk [8], namely, the determination of the number of clusters, cluster centers, and the assignment of the membership value. Our algorithm returns the number of cluster \( C \) and a set of membership values \( u_c(x_i) \) where \( i \) denotes the membership value of node \( i \) in cluster \( c \). The sum of membership value of a node \( i \) is 1, that is,

\[
\sum_{c=1}^{C} u_c(x_i) = 1
\]

2.1. Determining the cluster center
A cluster in protein interaction dataset is often characterized by a densely connected region on the graph. Based on [9], in this paper we define the density \( d_k \) of a node \( k \) as the ratio of number of edges between \( k \) and its neighbor, to the number of vertices in the neighbor of \( k \), which is

\[
d_k = \frac{|E_{N(k)\cup k}|}{|V_{N(k)\cup k}|}
\]

where \( E_k \) denotes a set of edges in the neighbor of \( k \), \( V_k \) denotes a set of nodes in the neighbor of \( k \), and \( N(k) \) denotes the set of the neighbors of \( k \), a set of nodes that is directly connected to the node \( k \). The search of cluster center is done iteratively. For each iteration, node with maximum density is selected as the cluster center and then removed along with its neighbor. The node removal is done to ensure that the cluster centers found are not too close to each other. If two disjoint region \( u \) and \( v \) of a graph \( G \) is indeed different cluster, then the node removal of one of the clusters would not affect the density of the other clusters significantly.

The iteration process is done until the maximum density is lower than a threshold \( \sigma \). Then the set of maximum density node of each iteration is defined as the cluster center for different clusters, and the number of cluster centers is the number of cluster. The default value of threshold parameter \( \sigma \) is 1. Parameter \( \sigma \) is optional, if it’s not provided, then the optimal number of clusters will be determined by the silhouette coefficient [10].
2.2. Spectral clustering
Basic concept of spectral clustering is to use the eigenvectors of the laplacian matrix of the input graph as the low-dimensional representation of the data [4,12,13,14]. Given the adjacency matrix $A_{ij}$ of the input graph, the laplacian matrix $L_{sym}$ is defined as

$$L_{sym} = I - D^{-1/2}AD^{-1/2}$$

where $D$ is diagonal matrix with its diagonal element $d_{ii} = \sum_i a_{ij}$. Now, given the number of clusters $k$, the spectral clustering algorithm computes the first $k$ eigenvectors (of $k$ smallest eigenvalues) of matrix $L_{sym}$, and assigns a $k$-dimensional coordinate vector to each node of graph $G$. This coordinate representation of the vertices is then clustered again using other clustering method (usually k-means clustering).

In summary, the steps done in spectral clustering algorithm is presented in the table below [12]

| Table 1. Spectral clustering algorithm |
|---------------------------------------|
| **Algorithm 1 : Spectral Clustering $(k)$** |
| - Input : Graph $G = (V,E)$ with $n$ nodes $1,2,\ldots,n$ and edge weight as pairwise similarity, number of clusters $k$ |
| - Form normalized laplacian matrix $L_{sym} = I - D^{-1/2}AD^{-1/2}$ from graph $G$ |
| - Compute the first $k$ eigenvectors $u_1,\ldots,u_k$ from matrix $L_{sym}$ |
| - Form matrix $U \in \mathbb{R}^{n \times k}$ with $u_1,\ldots,u_k$ as the columns |
| - For each $i = 1,\ldots,n$, set $y_i$ as the $i$ th row of matrix $U$. |
| - Cluster $(y_i)_{i=1,\ldots,n}$ using k-means algorithm resulting in $k$ clusters $C_1,C_2,\ldots,C_k$. |
| - Output: Cluster $A_1,A_2,\ldots,A_k$ where $A_i = \{j \mid y_j \in C_i\}$ |

Each data point from the $k$-dimensional data points $(y_i)_{i=1,\ldots,n}$ represents a node in the original graph. Fuzzy random walk uses these data points to assign a membership value for each node.

2.3. Fuzzy random walk
Fuzzy random walk consists of two steps, the computation of the auxiliary matrix and membership value assignment by random walk [8]. Consider the $k$-dimensional data points $(y_i)_{i=1,\ldots,n}$. First, we define a family of matrices $W(\beta)$ with its entries $w_{ij}(\beta)$, as

$$w_{ij}(\beta) = \begin{cases} \exp\left(-\frac{||y_i - y_j||^2}{\beta}\right), & \text{if node } i \text{ and } j \text{ connected in graph } G \\ 0, & \text{otherwise} \end{cases}$$

where $\beta$ is a positive real number and $\|\cdot\|$ denotes the euclidean distance. This entry value is defined to keep the distance between nodes in range $(0,1)$, where values near $1$ means a very close node pair, and values near $0$ would be a very far node pair. Now consider the function $L(\beta): (0,\infty) \to (0,\infty)$ defined by

$$L(\beta) = \sum_i \sum_j w_{ij}(\beta)$$

This function has two well-defined limit, that is

$$\lim_{\beta \to 0} L = 0 \text{ and } \lim_{\beta \to \infty} L = 2|E|$$

where $|E|$ is the number of edges of the graph. The selection of $\beta$ value is related to the time scale of the random walk. Small $\beta$ value corresponds to the case where almost no diffusion occur between
clusters, meanwhile large $\beta$ value corresponds to the case where all the nodes are highly connected, and thus diffusion occur instantaneously. We search for the intermediate value of $\beta$, when there is enough time to diffuse, but the equilibrium is not yet reached. We search for a $\beta^*$ value that satisfies

$$L(\beta^*) = 2\gamma_1 |E|$$

the adjustment of parameter $\gamma_1$ also gives the same meaning as $\beta$ but restricted to the range (0,1). The default value of $\gamma_1$ is 0.3. Then, we define the auxiliary matrix $W(\beta^*)$. Let $D = [d_{ii}]$ be a diagonal matrix with its diagonal entries defined by $d_{ii} = \sum_i w_{ij}$. We also define the transition matrix $P$, as

$$P = D^{-1}W$$

And the random walk time step, $\alpha$

$$\alpha = \left[ \frac{\gamma_2}{\lambda_2} \right]$$

where $\lambda_2$ denotes the second-smallest eigenvalue of matrix $P$ and $\gamma_2 \in (0, \infty)$ is internal parameter. Small value of $\gamma_2$ gives us a few time steps and large value of $\gamma_2$ gives us a large number of time steps. The default value of $\gamma_2$ is 0.1. Now, we assign the membership value of each point to each clusters. Let $S$ be a set of cluster center where $s_c$ denote the data point representation for a cluster center node of cluster $c$, we define the distance of a node $i$ to cluster $c$ as

$$\text{dist}(x_i, c) = (P^\alpha)_{s_c,i}$$

where $P^\alpha$ denotes the $\alpha$-step transition matrix and $s_c,i$ denotes the $s_c$-th row and $i$-th column of matrix $P^\alpha$. Finally, we define the membership value $u_c(x_i)$ of a node $i$ to cluster $c$, as

$$u_c(x_i) = \frac{\text{dist}(x_i, c)}{\sum_{c \in C} \text{dist}(x_i, c)}$$

This process is done to every node for each clusters. In summary, the steps done in our method, spectral clustering with fuzzy random walk is presented in table below

**Table 2. Spectral Clustering with Fuzzy Random Walk algorithm**

| Algorithm 2: Spectral Clustering with Fuzzy Random Walk ($\gamma_1, \gamma_2$) |
|---|
| **Input**: Graph $G = (V,E)$ with $n$ nodes 1,2, ..., $n$ and edge weight as pairwise similarity |
| For each node $v$, compute $d_v$ and search the maximum $d_v$. Let the corresponding node be $v_s$ |
| Remove $v_s$ and neighbor of $v_s$ from $G$, and repeat the search. Stop until the next $v_s$ found satisfies $d_s < \sigma$ |
| Set the number $v_s$ found as the number of cluster $k$, and then $v_s$ set as the cluster centers. |
| Form normalized laplacian matrix $L_{sym} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ from graph $G$ |
| Compute the first $k$ eigenvectors $u_1, ..., u_k$ from matrix $L_{sym}$ |
| Form matrix $U \in \mathbb{R}^{n \times k}$ with $u_1, ..., u_k$ as the columns |
| Set $(\gamma_i)_{i=1,...,n}$, the $i$ th row of matrix $U$ as the data points for fuzzy random walk |
| Define a family of matrices $W(\beta)$ with its entries defined above |
| Define a function $f(\beta) = L(\beta) - 2\gamma_1 |E|$ with $L(\beta) = \sum_i \sum_j w_{ij} (\beta)$ |
| Approximate the solution of $f(\beta) = 0$ using bisection method. Let the solution be $\beta^*$ |
| Compute the auxiliary matrix $W = W(\beta^*)$ and a diagonal matrix $D$ with $d_{ii} = \sum_j w_{ij}$ |
| Compute the transition matrix $P = D^{-1}W$ and its time step $\alpha = \left[ \frac{\gamma_2}{\lambda_2} \right]$ |
| Define $\text{dist}(x_i, c) = (P^\alpha)_{s_c,i}$ the distance of a node $i$ to cluster $c$ |
| Assign membership value $u_c(x_i) = \frac{\text{dist}(x_i, c)}{\sum_{c \in C} \text{dist}(x_i, c)}$ to each data point |
| **Output**: Membership value of each data points to each cluster.
3. Results and discussion

The implementation of our method is done on PPI data of TP53 protein using the python programming language on software Canopy 1.6.2 (64-bit). The program is run on computer that has specs: Intel® Core™ i5-4200U @ 1.60 GHz, 2.30 GHz, 4.00 GB RAM and Windows 7Ultimate operating system ©2009 Microsoft Corporate 64-bit.

The PPI dataset of TP53 protein that found in Homo sapiens is obtained from STRING (Search Tool for the Retrieval of Interacting Genes) version 10 [11] database available online on string-db.org. This interaction networks consists of 131 proteins (nodes) and 1478 interactions (edges) are presented in the figure below.

![PPI network of TP53 protein with 131 proteins and 1478 interactions](image)

**Figure 1.** PPI network of TP53 protein with 131 proteins and 1478 interactions

The cluster centers obtained from this PPI network consists of 6 proteins. The list of cluster centers, the protein name and density score is presented in table below.
Table 3. List of cluster centers

| Protein number | Protein Name | Density |
|---------------|--------------|---------|
| 18            | FZR1         | 13.7    |
| 127           | RPA1         | 7.2     |
| 0             | BCL2L1       | 2.8     |
| 58            | CREBBP       | 2       |
| 119           | TP53BP1      | 1.7     |
| 117           | TP53BP2      | 0.5     |

From this list we search for optimal number of clusters $k$ from $k=2$ to $k=6$ using the silhouette coefficient $s(i)$ of a node $i$ defined by [9]

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where $a(i)$ denotes the intra cluster average distance to node $i$ and $b(i)$ denotes the second-closest cluster average distance to node $i$. A silhouette coefficient score for a clustering result is the average of every node silhouette coefficient and is in range $(-1,1)$, where value close to 1 denotes a tightly grouped clusters with high dissimilarity between each clusters while value close to -1 denotes a clustering result that may not properly clustered. For this purpose we assign each node to a cluster based on the highest membership value and compute the silhouette coefficient score for $k=2$ to $k=6$ and set the top $k$ cluster centers to be the cluster center. The highest silhouette coefficient score is then chosen as the number of clusters. The figure below show the plot of silhouette coefficient score for $k=2$ to $k=6$

![Figure 2. Plot of Silhouette coefficient value to Number of Cluster](image)

From figure 2 we get the maximum silhouette coefficient value is obtained when the number of cluster is set to 5, and therefore we choose the number of clusters to be 5.

Even though the main protein, TP53 is not one of the cluster centers, we found that all cluster center listed in the previous table has a direct interaction with protein TP53. This indicates all the cluster formed has some connection regarding its function with protein TP53. The cluster centers for this clustering result are FZR1, RPA1, BCL2L1, CREBBP and TP53BP2. Also, from the membership value, we can infer that some protein has a greater importance than other protein in the same cluster. Although it not shown here, but our method gives the similar performance when compared with spectral clustering only. Without using fuzzy random walk, we can only infer the group in which each
node are in, but in our method, we could give more information on how this clusters are formed and the important component of each clusters. Figure 3 below shows the clustering result with contour map as membership value plot and node coloring is done based on maximum membership value of each node. The membership value gives the degree of membership of each node to each clusters, with darker color denotes larger value.

![Figure 3. Clustering Result with k = 5 with contour map as the membership value plot](image)

4. Conclusion and future work
Based on the analysis of our clustering results, we found our method on clustering TP53 PPI network produces a well-separated clusters with its membership value as an additional information on how these proteins interact with other clusters. The cluster centers found (FZR1, RPA1, BCL2L1, CREBBP and TP53BP2) are directly interacting with TP53 proteins, which indicates some connections with TP53 protein. It should be any further work to interpret the clustering result from our method to obtain more understanding by using data from the fields of biology and medical research.

For future work, we plan to study our method further not only to apply it to the biological dataset, but also a more general dataset in a graph form. In order to do that we would try to develop a more general method for detecting a cluster centers. Also, from computational approach, to achieve substantial performance over CPU approaches on a large dataset, we can use massively parallel computing approaches using GPU computing.

Acknowledgment
This research is supported by PITTA UI 2018 research grant from University of Indonesia and Directorate General of Higher Education Indonesia.

References
[1] Kleinberg J and Tardos É 2002 Approximation algorithms for classification problems with
pairwise relationships: metric labeling and Markov random fields. *J. ACM* **49** 616–39

[2] Duda RO, Hart PE and Stork DG 2001 *Pattern classification*. 2nd. Ed New York.

[3] Schaeffer SE 2007 Graph clustering *Comput. Sci. Rev.*

[4] Shi J and Malik J 2000 Normalized cuts and image segmentation *IEEE Trans. Pattern Anal. Mach. Intell.*

[5] de Las Rivas J and Fontanillo C 2010 Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks *PLoS Comput. Biol.* **6** 1–8

[6] Wang J, Li M, Deng Y and Pan Y 2010 Recent advances in clustering methods for protein interaction networks *BMC Genomics*

[7] Coifman RR, Kevrekidis IG, Lafon S, Maggioni M and Nadler B 2008 Diffusion Maps, Reduction Coordinates, and Low Dimensional Representation of Stochastic Systems *Multiscale Model. Simul.* **7** 842–64

[8] Cominetti O and Matzavinos A 2010 DifFUZZY: a fuzzy clustering algorithm for complex datasets *IJCIBSB, 1*(4):402 – 417

[9] Bader GD and Hogue CWV 2003 An automated method for finding molecular complexes in large protein interaction networks *BMC Bioinformatics* **4**

[10] Rousseeuw PJ 1987 Silhouettes: A graphical aid to the interpretation and validation of cluster analysis *J. Comput. Appl. Math.*

[11] Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, Kuhn M, Bork P, Jensen LJ and Von Mering C 2015 STRING v10: Protein-protein interaction networks, integrated over the tree of life *Nucleic Acids Res.*

[12] Ng AY, Jordan MI and Weiss Y 2002 On spectral clustering: Analysis and an algorithm *Adv. Neural Inf. Process. Syst. 14, Vols 1 2*

[13] Ding C, He, X, Zha H, Gu M and Simon H 2001 A min-max cut algorithm for graph partitioning and data clustering. In Proceedings of the first IEEE International Conference on Data Mining (ICDM)(pp. 107 – 114). Washington, DC, USA: IEEE Computer Society.

[14] Meila M and Shi J 2001 A random walks view of spectral segmentation*In8th International Workshop on Artificial Intelligence and Statistics (AISTATS).*