2013

Community Detection in Disease-Gene Network Based on Principal Component Analysis

Wei Liu  
Department of Computer Science, Information Science and Technology College, Yangzhou University, Yangzhou 225127, China

Ling Chen  
Department of Computer Science, Information Science and Technology College, Yangzhou University, Yangzhou 225127, China

Follow this and additional works at: https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology

Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Tsinghua Science and Technology by an authorized editor of Tsinghua University Press: Journals Publishing.
Community Detection in Disease-Gene Network Based on Principal Component Analysis

Wei Liu* and Ling Chen

Abstract: The identification of communities is imperative in the understanding of network structures and functions. Using community detection algorithms in biological networks, the community structure of biological networks can be determined, which is helpful in analyzing the topological structures and predicting the behaviors of biological networks. In this paper, we analyze the diseasome network using a new method called disease-gene network detecting algorithm based on principal component analysis, which can be used to investigate the connection between nodes within the same group. Experimental results on real-world networks have demonstrated that our algorithm is more efficient in detecting community structures when compared with other well-known results.

Keywords: disease-gene network; principal component analysis; community detection

1 Introduction

The completion of the human genome project has opened the doors to new research opportunities and challenges. One of the major goals of the post-genome era is to understand the role of genetics in human health and diseases. While fewer than 100 disease-gene associations were known before the project started in 1990, more than 1400 have been currently identified[1, 2]. Determining disease-gene associations will enhance the development of new techniques for prevention, diagnosis, and treatment of diseases. The “disease-gene associations” called “diseasome”[3] involves the construction of a bipartite network containing a set of diseases/phenotypes and a set of genes; here, if a gene causes a disease, they are interconnected. Using a single-graph theoretical framework, such a network can offer a platform to explore all known phenotypes and disease-gene associations, revealing the common genetic origins of several diseases. A diseasome bipartite network can be naturally divided into communities that can have the appearance of densely connected groups of vertices with sparser connections between groups (Fig. 1); such networks are often used to infer the genomic intervals that are associated with a disease under consideration.

In recent years, the detection and analysis of community structures have been intensively studied in relation to their applications in the analysis of networks[4]. Up to now, many algorithms have been

---

* Wei Liu and Ling Chen are with Department of Computer Science, Information Science and Technology College, Yangzhou University, Yangzhou 225127, China. E-mail: yzliuwei@126.com.

* To whom correspondence should be addressed.

Manuscript received: 2013-08-09; revised: 2013-08-23; accepted: 2013-08-26

---

Fig. 1 Vertices in several networks naturally fall into groups or communities: sets of vertices (shaded) within which there are several edges with only a smaller number of edges between the vertices of different groups.
proposed for community detection. Two such classical algorithms are the spectral bisection algorithm that employs the eigenvectors of the Laplacian matrix of a network\cite{5} and the Kernighan-Lin algorithm that improves the initial division by optimizing the number of within and between community edges\cite{6}. Recently, many algorithms based on modularity\cite{7} have been proposed. For example, Newman proposed a fast greedy algorithm\cite{8} to maximize the modularity. The same algorithm implemented with a better data structure has been proposed by Clauset et al.\cite{9}, which is essentially thousands of times faster than the Kernighan-Lin algorithm. An even faster and more accurate algorithm based on subgraph similarity has been proposed by Xiang et al.\cite{10} Ruan and Zhang\cite{11} proposed an efficient heuristic algorithm that combines spectral graph partitioning and local searching to optimize the modularity. Duch and Arenas\cite{12} presented a method to find the community structure by extremal optimization subject to the modularity. Wang et al.\cite{13} proposed a very fast algorithm for community detection based on local information. Newman proposed an algorithm using the eigenvectors of matrices\cite{14}. Chen et al.\cite{15} presented a fast and efficient algorithm by recursively adding a node into a partial community until a local optimal community was obtained.

However, most of the previous works are applicable for unipartite networks comprising only one type of node. In real-world situations, there are many bipartite networks composed of two types of nonoverlapping nodes as well as links that have at least one end of a node in each set. One of the examples of such networks is the disease-gene network. In this paper, a new approach named Disease-Gene Network detecting algorithm based on Principal Component Analysis (DGN,PCA) is proposed to identify the communities in disease-ome bipartite networks. The main idea is to initially transform the bipartite network into an equivalent or linear graph; then, the graph’s incidence matrix needs to be improved, followed by the principal component analysis of the incidence matrix. Finally, communities in two parts of the bipartite network are detected. A real-world network, namely, a disease-gene network, is used to test the performance of our algorithm. The experimental results have revealed that our algorithm is more efficient and practicable.

2 Method

2.1 Basic properties of bipartite networks

A bipartite network \( G = (V, E) \) is a graph whose vertices can be divided into two disjoint sets \( V_1 \) and \( V_2 \) such that every edge connects a vertex in \( V_1 \) to a vertex in \( V_2 \); that is, \( V_1 \) and \( V_2 \) are independent sets. Besides, there is no edge connected between any two vertices within \( V_1 \) or \( V_2 \). Suppose the partitions of a bipartite network have the sizes \( |V_1| = n_1 \) and \( |V_2| = n_2 \). The adjacency matrix of a bipartite network can be represented in the form of a block matrix. In this adjacency matrix, if the first \( n_1 \) rows and \( n_1 \) columns correspond to the \( n_1 \) vertices in \( V_1 \) and the first \( n_2 \) rows and \( n_2 \) columns correspond to the \( n_2 \) vertices in \( V_2 \), the adjacency matrix \( A \) can be defined as

\[
A = \begin{bmatrix}
0 & A_1 \\
A_2 & 0
\end{bmatrix},
\]

where \( A_1 \) is an \( n_1 \times n_2 \) matrix and \( A_2 \) is an \( n_2 \times n_1 \) matrix. Furthermore, \( A_1 \) and \( A_2 \) satisfy the following condition: \( A_1 = A_2^T \).

Example 1: Consider the bipartite network \( H' \) as shown in Fig. 2, we can obtain its adjacency matrix as follows:

\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}.
\]

We set \( A' = \begin{bmatrix}
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1
\end{bmatrix}, \) and \( A = \begin{bmatrix}
0 & A'^T \\
A' & 0
\end{bmatrix} \). Note that the matrix \( A' \) uniquely

![Fig. 2 Bipartite network \( H' \).](image-url)
represents the bipartite network.

2.2 Linear graph

For any graph (such as the one shown in Fig. 3), we assume its edges to be a set of vertices \( V_1 \) and its vertices as another set of vertices \( V_2 \), which collectively constitute a bipartite network. If a vertex of \( V_2 \) is related to a vertex of \( V_1 \), an edge will be added between the two vertices. After this conversion, graph \( G \) can be described as a bipartite network \( H' \) shown in Fig. 2.

As noted earlier, the bipartite network \( H' \) has preserved all the information of the original graph \( G \). Conversely, \( H' \) can be converted into \( G \) with all the information.

It can be seen that the submatrix \( A' \) of the above mentioned adjacency matrix \( A \) is the incidence matrix of \( G \), as shown in Fig. 3; this indicates that Fig. 3 reserves all the information of \( H' \).

If we assume \( A, B, C, \) and \( D \) in Fig. 2 as a set of vertices and \( E_1, E_2, \) and \( E_3 \) as a set of edges, we would obtain a linear graph corresponding to graph \( G \), as shown in Fig. 4.

In accordance with Fig. 4, it is evident that the submatrix \( A'^T \) of the adjacency matrix \( A \) is the incidence matrix of the linear graph \( G' \). Therefore, it can be concluded that \( G' \) has retained all the information of the original graph \( G \) and the bipartite network \( H' \). Further, each vertex in \( G \) would correspond to a clique in \( G' \), i.e., a complete subgraph.

2.3 Basic idea of our algorithm

Given a bipartite network, such as the gene-disease bipartite network \( H \) shown in Fig. 5, we can view the nodes of the disease phenome as vertices and the nodes of the disease genome as edges, collectively constituting graph \( G \). After clustering the vertices of \( G \), we can detect its communities by the use of community detection algorithms, followed by clustering for disease phenomes.

Similarly, we can also consider the nodes of the disease genome as the vertices and the nodes of the disease phenome as edges, collectively constituting a linear graph \( G' \) of \( G \). After mining the communities for the vertices of \( G' \), the clustering for the disease genome can be undertaken.

However, as mentioned above, the bipartite network \( H \), graph \( G \), and linear graph \( G' \) are equivalent. Therefore, we have to only make clustering for the vertices of \( G \) or \( G' \), i.e., clustering for the corresponding edges. In other words, as long as the community detection for \( G \) or \( G' \) is conducted, we can simultaneously obtain the clustering results for the disease phenome as well as the disease genome, and therefore, additional clustering for the disease phenome and disease genome is not required.
3 DGN, PCA

Suppose that the graph $G$ is generated by the bipartite network $H$ whose adjacency matrix is $A = \begin{bmatrix} 0 & A^T \\ A' & 0 \end{bmatrix}$, and the incidence matrix of $G$ is $A'$. We use the PCA on the vectors of $A'$ for detecting the communities in $G$ based on the incidence matrix.

3.1 PCA

PCA is a mathematical procedure invented by Karl Pearson in 1901. Now it is mostly used as a tool in dimension reduction, feature extraction, and constructing predictive models. PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called the principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a manner that the first principal component has the highest possible variance (that is, it accounts for as much of the variability in the data as possible), and each succeeding component, in turn, has the highest possible variance (that is, it accounts for as much of the variability in the data as possible), under the constraint that it is orthogonal to (uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. Depending on the field of application, it is also called the discrete Karhunen-Loève Transform (KLT), the Hotelling transform, or Proper Orthogonal Decomposition (POD).

Assume that there are data points $x_1, x_2, \ldots, x_m$ in an $n$-dimensional space, which can be denoted by the following matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

where $\bar{x}_j = \frac{1}{m} \sum_{i=1}^{m} x_{ij}$ and $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$.

Let the matrix be denoted as

$$\hat{x} = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1n} - \bar{x}_n \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2n} - \bar{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} - \bar{x}_1 & x_{m2} - \bar{x}_2 & \cdots & x_{mn} - \bar{x}_n \end{bmatrix}.$$

Therefore, the covariance matrix is

$$S = \frac{1}{m-1} \hat{x}\hat{x}^T = \frac{1}{m-1} \begin{bmatrix} \sum_{k=1}^{m} (x_{1k} - \bar{x}_k)(x_{1k} - \bar{x}_k) & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \sum_{k=1}^{m} (x_{mk} - \bar{x}_k)(x_{1k} - \bar{k}) & \cdots & \cdots \end{bmatrix},$$

where $S$ is an $m \times m$ symmetrical matrix and it can be rewritten in the form of a matrix-vector:

$$S = \frac{1}{m-1} \sum_{k=1}^{m} (x_k - \bar{x})(x_k - \bar{x})^T.$$

Next, the eigenvalues of $S$, namely, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_m > 0$, and the corresponding eigenvectors, $l_1, l_2, \ldots, l_m$, are computed; this process is called orthogonalization, that is, $l_i \cdot l_i^T = 1$ and $l_i \cdot l_j^T = 0$.

We select $p$ bigger eigenvalues, namely, $\lambda_1, \lambda_2, \ldots, \lambda_p$ whose corresponding eigenvectors are $l_1, l_2, \ldots, l_p$. Given an $m \times p$ matrix comprising $u = (l_1, l_2, \ldots, l_p)$, the data $x$ (an $m$-dimensional vector) can be denoted as $x'$ in the new space such that $x' = x \cdot u$, where $x'$ is a $p$-dimensional vector.

3.2 PCA of incidence matrix $A'$

To simplify the analysis of the principal components, we have to reconstruct the incidence matrix as follows: (1) When the number of “1”s in each column is greater than 2 (normally, in the incidence matrix, “1”s only appear twice in each column, but there may be multiple “1”s determined by the properties of the bipartite network), the “1”s in each column are changed into

$$\frac{1}{n}$$

and their pair-wise combination is created. After these transformations, we obtain $C_k^2$ columns, for instance:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

will be converted into

$$\begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 1 & 0 \\ 3 & 1 \\ 1 & 3 \\
3 & 2 \end{bmatrix},$$

which is then continuously changed into
According to the above, we can obtain the eigenvector as the discriminative vector.

(3) The minimal eigenvalue of $S$ is 0.

(4) Suppose that $G$ is generated by the incidence matrix $A_1$ of the bipartite network $H$ and the covariance matrix $S$ and the Laplacian matrix of $G$ are equal only up to a constant factor.

Proof $S = \frac{1}{N-1}A''A'^T$, and its element $s_{ij} = \sum_{k=1}^{n}a_{ik}a_{jk}/N - 1$.

On the assumption that the adjacency matrix of $H$ is $A = \begin{bmatrix} 0 & A_1^T \\ A_1 & 0 \end{bmatrix}$, the incidence matrix $A_1$ generates the graph $G$ whose adjacency matrix is $B = [b_{ij}]$. Assume that the degree of the vertex $i$ is $d_i$. When $i \neq j$, $S_{ij} = \frac{1}{N-1}\sum_{k=1}^{n}a_{ik}a_{jk} = -2b_{ij}/N - 1$; otherwise, when $i = j$, $S_{ii} = S_{i_i} = \frac{1}{N-1}\sum_{k=1}^{n}a_{ik}a_{ik} = 2d_i/N - 1$. Therefore, it can be seen that $S_{ij} = 2(d_i - b_{ij})/N - 1$. If the constant coefficient $\frac{2}{N-1}$ is omitted, $S$ is equal to the Laplacian matrix of $G$.

4 Algorithm Framework

Based on the above mentioned analysis, we can obtain the framework of our algorithm as Algorithm 1.

5 Experimental Results and Analysis

In this section, we conduct a set of experiments to compare the performance of the DGN PCA algorithm with other typical algorithms. All the experiments were conducted using a computer with a 3.0 GHz Pentium processor with 2 GB memory. All the codes were compiled using MATLAB 7.0.

Now, we employ the disease-gene bipartite network a real-world dataset to test our algorithm. The diseasome construction contains a set of diseases/phenotypes and a set of genes; here, if a gene causes a disease, they are interconnected. Using a single-graph theoretical framework, such a network can offer a platform to explore all known phenotypes and disease-gene associations, revealing the common genetic origins of several diseases.

We extract a small subset from the Online Mendelian
Algorithm 1 DGN_PCA:

Input: Given a bipartite network $W$, its adjacency matrix $A = \begin{bmatrix} 0 & A_1^T \\ A_1 & 0 \end{bmatrix}$. Two sets of vertices are defined in $W$, namely, $V_1$ and $V_2$ whose sizes are $n_1$ and $n_2$, respectively.

Output: Community groups of $V_1$ and $V_2$.

Begin
  1. if $n_1 > n_2$, $A_2 = A_1$, else $A_2 = A_1^T$;
  2. Reconstruct $A_2$ and obtaining the matrix $A_3$;
  3. Computing $S = A_3A_1^T$;
  4. Calculating the eigenvalues of $S$, namely, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and the corresponding eigenvectors, namely, $u_1, u_2, \cdots, u_n$;
  5. Eliminating the minimal eigenvalue $\lambda_1 = 0$;
  6. for $i = 2$ to $n_1 - 1$, do $l_i = \lambda_i + 1 - \lambda_i$; end for
  7. Searching for the maximal $l_i$ denoted as $l_k$ among $l_i, \cdots, l_{n_1-1}$ and then choosing $u_2, u_3, \cdots, u_k$ to build an $n \times (k-1)$ matrix, that is, $u = [u_2, u_3, \cdots, u_k]$;
  8. Supposed the row vector of $u$ is $p_1, p_2, \ldots, p_n$, namely, 
     \[
     u = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}; \text{ use the k-means algorithm for the clustering of } p_1, p_2, \ldots, p_n;
     \]
  9. Hence, the clustering of the vertices and the corresponding edges can be performed.

Inheritance in Man (OMIM) dataset and construct a diseaseome bipartite network named $H$, as shown in Fig. 5. It can be seen that there are 19 hereditary diseases, 19 disease genes, and associated linkages in $H$, where the circles and rectangles correspond to disorders and disease genes, respectively. A link is placed between a disorder and a disease gene if the mutations in that gene lead to the specific disorder. Based on Fig. 5, mutual influences and inner relationships between these disorders and disease genes can be studied, which can then be extended to all the known disease genes and hereditary disorders for obtaining general conclusions. We apply our algorithm to community detection in diseaseome networks and investigate the correlations between the elements of the disease phenome and disease genome.

To make further analysis for convenience, we compare our algorithm with the algorithm in Ref. [16]. The results are shown in Figs. 6-9.

From Figs. 6 and 7, it is evident that our algorithm yields seven modules regarding the disease phenome and three modules regarding the disease genome, while Ref. [16] yields a different result. The colors of the circles and rectangles correspond to the community to which the disorders or disease genes belong.

In accordance with the visual representation of the disease-gene network, it can be concluded that our results are far better than those obtained by Ref. [16], which indicates that our algorithm can comprehensively reveal the relationships between the diseases and

Fig. 6 Partitions of the disease-gene network obtained using our method.

Fig. 7 Partitions of the disease-gene network obtained by Ref. [16].
disease genes; this can be helpful in studying the pathogenesis of inherited diseases and can assist in gene diagnosis and therapy.

We also perform the clustering of two projections of the diseasome: the projection on diseases called the Human Disease Network (HDN, left-hand side of Fig. 5) and the projection on genes called the Disease Gene Network (DGN, right-hand side of Fig. 5). The comparative results of the matching rates of the two algorithms are shown in Fig. 8 and Fig. 9.

According to Fig. 8 and Fig. 9, it is evident that in addition to the increases in precision, regardless of clustering being performed for HDN or DGN, our algorithm yields higher matching rates, indicating greater efficiency.

Based on Figs. 6-9, all the results demonstrate that disease occurrence is not isolated but relevant to several other diseases. The probability of the disorder in the same community induced by each other is greater than those with no connection. The results also demonstrate that genes associated with similar disorders show both higher likelihood of physical interactions between their products and higher expression profiling similarity for their transcripts. Further, this distinctly reveals the common genetic origin of several diseases, which is useful in the research of disease prediction and the development of new drugs.

6 Conclusions

In this paper, we propose a novel community detection method called DGN_PCA based on the principal component analysis and successfully apply it to a real-world network, namely, the disease-gene network. We convert the original bipartite network into an equivalent graph or linear graph and reconstruct the incidence matrix of the graph; communities are then detected using the PCA method. The experimental results reveal that our algorithm can not only successfully identify the modular structure but also detect more communities of biological significance, which are essential in disease prevention and medical diagnosis.

Acknowledgements

This research was supported in part by the Natural Science Foundation of Education Department of Jiangsu Province (No. 12KJB520019), the National Science Foundation of Jiangsu Province (No. BK20130452), Science and Technology Innovation Foundation of Yangzhou University (No. 2012CXJ026), the National Natural Science Foundation of China (Nos. 61070047, 61070133, and 61003180) and the National Key Basic Research and Development (973) Program of China (No. 2012CB316003).

References


Wei Liu received her BS degree and MS degree in computer science from Yangzhou University, China in 2004 and 2007, respectively. In 2010, she received her PhD degree from Nanjing University of Aeronautics and Astronautics. She is currently an associate professor in the Institute of Information Science and Technology, Yangzhou University, China. Her research interest includes data mining, bioinformatics, and parallel processing. She has published more than 30 papers in journals and conferences.

Ling Chen received his BS degree from Yangzhou Teachers College. Currently he is a professor of the Information Technology College, Yangzhou University. He is a senior member of the Chinese Computer Society, member of IEEE and ACM Computer Society, vice director of Jiangsu Computer Society, also the director of Yangzhou Computer Society. His research interests include artificial intelligence, data mining, system optimization, and complex network analysis.