Community Detection Methods using Eigenvectors of Matrices

Yan Zhang

Abstract

In this paper we investigate the problem of detecting communities in graphs. We use the eigenvectors of the graph Laplacian in order to solve a traditional graph partitioning problem. Then we will examine the concept of the modularity matrix, which is used to measure the deviation between the actual and expected number of edges between each pair of vertices, and how its eigenvectors can be used to construct the communities in a network. Finally, we will review some applications of these concepts to real-world networks.

1 Introduction

A graph or a network, defined as a set of nodes connected to each other by links, serves as an integral structure in many disciplines, including mathematics, computer science, physics, biology, and the social sciences. An interesting problem in network theory is that of finding *community structure* in a network, which aims to find groups of nodes, or *communities*, which have more links within the communities than between communities.

There is typically useful information found at the community level that is not readily available from studying the network as a whole. For example, consider the United States Congress as a network of the Congressmen. Then the application of community-detection methods will typically separate the Republicans and Democrats into different communities. If we find data concerning these two communities, such as the average vertex degree, we can draw conclusions concerning members of the two parties, which could not be done by looking at the statistics for the whole graph. In addition, we can also determine how strongly each node is tied to its community. This is useful because nodes that are only weakly connected to their own community may serve as 'bridges' between different communities [5]. Such a node may have a useful interpretation; for example, in the Congressional network the node may be interpreted as a moderate.

In the last few years, many methods of community detection have been developed [1], but we will focus on a few in particular. The traditional method of spectral partitioning, which uses the graph Laplacian matrix, is first discussed along with its limitations. In the next section, the concept of modularity, a

benefit function which measures the difference between the number of links within a community versus the expected number of links, is formulated as a matrix in order to give a method for community detection. We finish the paper with a discussion of some applications of the method.

2 Spectral partitioning of graphs

Consider an undirected, simple graph G with n vertices $\{1, 2, \ldots, n\}$ and m edges, and let $A = (a_{ij})$ be its adjacency matrix, where $a_{ij} = 1$ if there is an edge connecting vertices i and j, and $a_{ij} = 0$ otherwise. Let $k_i = \sum_j A_{ij}$ be the degree of vertex i, and let D be a diagonal matrix with $d_{ii} = k_i$. Then we define the *Laplacian matrix* of G to be $L := D - A$. Two useful properties of the Laplacian matrix are that it is positive semidefinite [7], and that 0 is an eigenvalue of L with eigenvector $(1, 1, \ldots, 1)$, because its rows and columns sum to 0. Define a *vertex separator* S to be a set of vertices such that $G \backslash S$ consists of two connected components. We now introduce an algorithm from [7] which will find a vertex separator S that contains as few vertices as possible and separates the vertices of G into nearly equal parts.

Algorithm 1

1. Compute the eigenvector v_2 corresponding to the second smallest eigenvalue, which is the first non-zero eigenvalue, and let v_m be the median value the components of v_2.

2. Partition the vertex set into two sets A' and B', where $A' = \{i : v_i \leq v_m\}$, and $B' = V \backslash A'$.

3. Let A_1 be the set of vertices in A' adjacent to some vertex in B', and let B_1 be the set of vertices in B' adjacent to some vertex in A'. Now compute H, the bipartite subgraph induced by the vertex sets A' and B'.

4. Now we find the minimal vertex cover S of H by taking a maximum matching [3]. Choose $A_s \subset A_1$ and $B_s \subset B_1$ such that $S = A_s \cup B_s$. Then S is the vertex separator we want, and it separates G into 2 parts with vertex sets $A' \backslash A_s$ and $B' \backslash B_s$.

This algorithm separates a graph into two components. In addition, since only $|A_s| + |B_s|$ is invariant, typically the structure of H permits some freedom in choosing A_s and B_s,, so we can choose them such that A and B are less unequal in size. The algorithm has a worst case time complexity of $O(\sqrt{n}m)$. Unfortunately, this means that the method is only reasonable for sparse networks, or networks which have relatively few edges, and does not allow for weighted networks, which are common in social networks. In addition, sometimes it is desirable instead to find a partition of the graph that may not disconnect the graph completely, but leaves as few edges as possible between the two parts. To do this, we introduce a result from [5].

Let $\mathbf{s}$ be an index vector of the groups, that is, let $s_i = 1$ if i is one group and $s_i = -1$ if i is in the other group. We wish to compute the cut size R, which is the number of edges that run between the two groups. R can be computed by the equation

$$R = \frac{1}{2} \sum_{i,j:s_i \neq s_j} A_{ij}. \tag{1}$$

Our objective is to minimize this quantity. First we consider the problem if $\mathbf{s}$ could take on any value rather than simply ± 1.

Lemma 1. *If the restriction that* $\mathbf{s}$ *must take values of* ± 1 *is ignored, then the vector* $\mathbf{s}$ *that minimizes* R *will be the eigenvector corresponding to the smallest eigenvalue of the Laplacian Matrix.*

Proof. By the definition of $\mathbf{s}$, we can rewrite (1) as

$$R = \frac{1}{4} \sum_{ij} (1 - s_i s_j) A_{ij}. \tag{2}$$

Now using the definition of the degree of the vertex, we can express the first term of the sum in (2) as

$$\sum_{ij} A_{ij} = \sum_i k_i = \sum_i s_i^2 k_i = \sum_{ij} s_i s_j k_i \delta_{ij},$$

where δ_{ij} is the Kronecker delta function. Therefore,

$$R = \frac{1}{4} \sum_{ij} s_i s_j (k_i \delta_{ij} - A_{ij}), \tag{3}$$

but note that $k_i \delta_{ij} - A_{ij}$ is L_{ij}, so (3) has a convenient matrix form

$$R = \frac{1}{4} \mathbf{s}^T L \mathbf{s}.$$

Now write $\mathbf{s}$ as a linear combination of the normalized eigenvectors $\mathbf{v}_i$ of L by taking $\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v_i}$, where $a_i = \mathbf{v_i}^T \mathbf{s}$. This in combination with $\mathbf{s}^T \mathbf{s} = n$ means that

$$\sum_{i=1}^{n} a_i^2 = n.$$

Therefore, we now have

$$R = \sum_i a_i \mathbf{v_i}^T L \sum_j a_j \mathbf{v_j} = \sum_{ij} a_i a_j \lambda_j \delta_{ij} = \sum_i a_i^2 \lambda_i, \tag{4}$$

where λ_i are the eigenvalues of L, written in increasing order. Now it is clear that in order to minimize R, we wish to choose the a_i such that the greatest possible weight is assigned to the eigenvectors coresponding to the smallest eigenvalues. In particular, if we choose $a_1 = 1$ and $a_j = 0$ for all $j \neq 1$, we will have the minimal R.

□

We know that smallest eigenvalue of the Laplacian matrix is 0 and its corresponding eigenvector is $(1, 1, \ldots, 1)$. However, if we choose this vector to be the index vector, then we place all of the vertices in one group and create a cut size of 0. Of course, this is not an interesting solution to the problem, so we would like to find a nontrivial solution. In this case, if we fix the sizes of the two groups to be n_1 and n_2, then we can fix the coefficient a_1 in (4) to be

$$a_1^2 = (\mathbf{v_1}^T \mathbf{s})^2 = \frac{(n_1 - n_2)^2}{n}. \tag{5}$$

Then if we wish to minimize R, assuming that there are no more constraints on $\mathbf{s}$, we should take $\mathbf{s}$ parallel to the second eigenvector $\mathbf{v_2}$, also called the *Fiedler vector*, but $\mathbf{s}$ is constrained to take values of 1 or -1, and the number of vertices in each group is contrained. Therefore, the approximation is not perfect. In addition, constraining the number of vertices in each group is in many instances an unrealistic simplication of the problem, because in most cases we do not know the sizes of the communities in advance [5]. Therefore, we need a different approach to more capably handle these cases. The approach that will be discussed is based on the modularity matrix.

3 The Modularity Matrix

When considering the problem of finding communities in a network, we note that communities should intuitively have more links between them than what is expected. This is embodied by the concept of *modularity* [5], [6]. The modularity Q of a set of communities is defined as

$$Q = (\text{number of edges within communities}) - (\text{expected number of edges within communities}).$$

Our objective is to find a partition of the network into communities such that the modularity is maximized. First we consider the problem of the optimal division of a network into two subcommunities. We will formulate the modularity in matrix form [5]. Consider an *adjacency matrix* A, and a "null model" matrix P that gives the expected number of edges between any two vertices. The modularity is then

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - P_{ij}] \delta(g_i, g_j), \tag{6}$$

where m is the total number of edges in the network, g_i is the community in which i is contained, and δ is the Kronecker delta function.

To proceed, one must find a suitable P. First, we assume that P is symmetric and has zero diagonal, since the graph is undirected and simple, and that the sum of the entries in P are the same as the sum of the entries in A. That is

$$\sum_{ij} P_{ij} = \sum_{ij} A_{ij}. \tag{7}$$

Second, we assume the "null model" has the same degree distribution as the real-world network. Therefore, we require that

$$\sum_{ij} P_{ij} = k_i, \tag{8}$$

where k_i, the degree of vertex i, is calculated by using

$$k_i = \sum_j A_{ij}.$$

Clearly, if (8) is satisfied, then (7) is automatically satisfied as well. Now if we make a simple assumption of randomness, we can compute P_{ij}, as demonstrated by the next lemma.

Lemma 2. *If we place edges subject to* (8) *uniformly at random, then*

$$P_{ij} = \frac{k_i k_j}{2m}.$$

Proof. If edges are placed at random, then the probability that a randomly chosen edge is attached to the vertex i depends only on the degree k_i of that vertex, and the probability that the edge is attached to some vertex at one end is independent of the probability that it is attached to any other vertex at the other end. Therefore, the expected number of edges P_{ij} between vertices i and j is $f(k_i)f(k_j)$, where f is the same function for both i and j because P_{ij} is symmetric. Then (8) implies that

$$\sum_{j=1}^{n} P_{ij} = f(k_i) \sum_{j=1}^{n} f(k_j) = k_i,$$

for all i, which means that $f(k_i) = Ck_i$ for some constant C. Now if we apply (7), we get that

$$2m = \sum_{ij} P_{ij} = C^2 \sum_{ij} k_i k_j = (2mC)^2,$$

which means that $C = \frac{1}{\sqrt{2m}}$, so

$$P_{ij} = \frac{k_i k_j}{2m}.$$

□

Because we are considering the division of a network into two subcommunities, we can define an index vector $\mathbf{s}$ with $|\mathbf{s}| = n$, and each entry has the value 1 if the corresponding node is in one community and the value -1 if it is in the other. We can now express the Kronecker delta in (6) in terms of the elements of $\mathbf{s}$ by noting that

$$\delta(g_i, g_j) = \frac{s_i s_j + 1}{2},$$

which means that we can manipulate (6) to get

$$Q = \frac{1}{4m}\sum_{ij}[A_{ij} - P_{ij}](s_i s_j + 1) = \frac{1}{4m}\sum_{ij}[A_{ij} - P_{ij}]s_i s_j$$

where the second equality follows due to (7). This expression can be written in the matrix form

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{9}$$

where $\mathbf{B} := A - P$ is the *modularity matrix*.

If $\mathbf{s}$ were unconstrained, then the modularity would be maximized for $\mathbf{s}$ parallel to $\mathbf{u}_1$, the eigenvector of B with the largest eigenvalue (i.e., the leading eigenvector). However, the elements of $\mathbf{s}$ must be ± 1, so we must settle for an approximation. We want $\mathbf{s}$ to be as close to $\mathbf{u}_1$ as possible by assigning the value $+1$ to an element in $\mathbf{s}_1$ if the corresponding element in $\mathbf{u}_1$ is positive and assigning the value -1 if the corresponding element is negative. If the corresponding element in $\mathbf{u}_1$ is 0, we assign the element to the group that would give the greatest modularity. This yields a division of the network into two subcommunities which maximizes the modularity.

We would like to apply this procedure recursively in order to divide the network into more than 2 subcommunities, but the contribution to the modularity for each subdivision after the first is not given correctly by the preceding formulas. Therefore, we need to find the modularity contribution from dividing a community. Let G be the set of vertices in the community to be divided, and let n_G be the number of vertices within the community. Let $\mathbf{S}$ be an $n_G \times c$ index matrix denoting the subdivision of the community into c subdivisions such that $S_{ij} = 1$ if vertex i belongs to subcommunity j and $S_{ij} = 0$ otherwise. Therefore, we can now find the difference ΔQ between the modularities of the network before and after the subdivision of the community by writing

$$\begin{aligned}
\Delta Q &= \sum_{i,j\in G}\sum_{k=1}^{c} B_{ij}S_{ik}S_{jk} - \sum_{i,j\in G} B_{ij} \\
&= \sum_{k=1}^{c}\sum_{i,j\in G}\left[B_{ij} - \delta_{ij}\sum_{l\in G} B_{il}\right]S_{ik}S_{jk} \\
&= \mathrm{Tr}(\mathbf{S}^T\mathbf{B}^{(G)}\mathbf{S}),
\end{aligned} \tag{10}$$

where the constant factor $\frac{1}{4m}$ has been omitted for convenience, because it does not affect $\mathbf{S}$.

$\mathbf{B}^{(G)}$ is an $n_G \times n_G$ generalized modularity matrix indexed by the vertex labels i, j of the vertices in G and having values

$$B_{ij}^{(G)} = B_{ij} - \delta_{ij}\sum_{l\in G} B_{il},$$

where the B_{ij} are the values of $\mathbf{B}$. Note that the form of (10) is the same as that of (9), with $\mathbf{s}$ replaced by $\mathbf{S}$, and that if G consists of all vertices of

the graph, $\mathbf{B}^{(G)} = \mathbf{B}$. Therefore, substituting $\mathbf{B}^{(G)}$ in place of $\mathbf{B}$ gives the correct modularity change in the network for the division of any subcommunity, including the entire network. An algorithm for computing the division of a network into more than 2 communities can now be given. We recursively apply the procedure to the 2 communities created at each step by recomputing $\mathbf{B}^{(G)}$ for the subcommunities, and we terminate the algorithm if no division with positive modularity change can be found [5].

However, in some cases this procedure gives very few iterations. For example, in the legislation cosponsorship network of Congress, which I studied this past summer for my SURF, the procedure terminates after one division. Since we would like to see more hierarchical structure, we need to adapt the procedure. Instead of terminating the procedure when no division with positive modularity can be found, we can try to find a division by considering the community as a full network. This means that we use $\mathbf{B}$ restricted to the vertices of the community instead of computing $\mathbf{B}^{(G)}$, which is expressed formally in the following algorithm.

Algorithm 2

1. Compute the altered modularity matrix $\mathbf{B}^{(G)}$ of the community G; we know that $\mathbf{B}^{(G)} = \mathbf{B}$ if G is the entire network.

2. Compute the leading eigenvector of $\mathbf{B}^{(G)}$. Use it to compute the index vector $\mathbf{s}$ and the modularity change Q.

3. If $Q > 0$, then apply the procedure on the two communities created by the partition as indicated by $\mathbf{s}$.

4. If $Q = 0$, then take as the modularity matrix $\mathbf{B}'$,the submatrix of $\mathbf{B}$ corresponding to the vertices of G, and compute the leading eigenvector, index vector $\mathbf{s}'$, and modularity change Q'. Note that by construction 0 is always an eigenvalue of the modularity matrix.

5. If $Q' > 0$, then apply the procedure on the two communities created by the partition as indicated by $\mathbf{s}'$, using $\mathbf{B}'$ as the modularity matrix of the full network.

6. If $Q' = 0$, then terminate the algorithm.

4 Applications

The original Newman algorithm has been applied to several networks with reasonable success [6]. For example, the algorithm was applied to the social network of 62 bottleneck dolphins living in Doubtful Sound, New Zealand [4]. When labelling the dolphins with the known group identifications that occurred after a dolphin left the group, the Newman algorithm identified more dolphins correctly than the spectral partitioning algorithm did; it misidentified only 3 out of the 62 dolphins while the spectral partitioning algorithm misidentified 10 dolphins

[5]. The algorithm has also been used to analyze a network of political books [2], which was compiled by taking recent books concerning American politics and connecting them with edges based on data from Amazon.com indicating which pairs of books are frequently purchased by the same customers. In addition to separating books aligned with conservative or liberal viewpoints, as discussed in [6] the elements of the first eigenvector $\mathbf{u_1}$ can be used as an indicator of the strength of connection of each element to its community. Therefore, we can pick out a most left-wing and right-wing book: in this case they turn out to be *Bushwacked* by Molly Ivins and Lou Dubose and *A National Party No More* by Zell Miller, respectively [5].

The altered Newman algorithm has also been applied to several applications. For example, it has been applied to the Zachary Karate Club network, studied by Zachary in 1977 [10], which is a karate club at American University that split into two karate organizations. This network has become a canonical test case for community detection methods. Using the modified Newman algorithm predicts the split correctly by correctly identifying the association of each member. In addition, it has been applied to the legislation cosponsorship network of the US Congress [11]. This network uses the members of Congress as nodes and connects them with links if they have sponsored or cosponsored a bill together. In fact, the need to modify the Newman algorithm came from this network, because applying the original algorithm to it yielded only one division, which only told us the well-known political truth that the most important predictor of a politician's behavior is his party affiliation. My advisor and I devised the modification during this past summer in order to find more structure in the network. With the modified algorithm, we were able to get additional divisions of the network and thus additional hierarchical levels, which uncovered additional features , such as a group of Southern Democrats that seem to have collaborated with the Republicans, especially in the 1970's, or a group of Northeastern Republicans which seem to be collaborating with the Democrats.

References

[1] Danon, L., Dáz-Guilera, A., Duch, J., and Arenas, A. Comparing community structure identification. **2005**. *J. Stat Mech* **P09008**.

[2] Krebs, V. Unpublished.

[3] Lawler, E. L. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart, and Winston: 1976.

[4] Lusseau, D., Schneider K., Boisseau O. J., Haase P., Slooten E., and Dawson S. M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Can geographic isolation explain this unique trait? **2003**. *Behavioral Ecology and Sociobiology* **54**, 396-405.

[5] Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. **2006**. *Physical Review E* **74**, 036104.

[6] Newman, M. E. J. Modularity and community structure in networks. **2006**. *Proceedings of the National Academy of Sciences* **103**(23), 8577–8582.

[7] Pothen, A., Simon, H., and Liou, K.-P. Partitioning sparse matrices with eigenvectors of graphs. **1990**. *SIAM J. Matrix Anal. Appl.* **11**, 430-452.

[8] Porter, M. A., Mucha, P. J., Newman, M. E. J., & Warbrand, C. M. A network analysis of committees in the United States House of Representatives. **2005**. *Proceedings of the National Academy of Sciences* **102**(20), 7057–7062.

[9] Porter, M. A, Mucha, P. J, Newman, M. E. J, & Friend, A. J. Community Structure in the United States House of Representatives. **2006**. arXiv.org:physics/0602033.

[10] Zachary, W. W. An Information Flow Model for Conflict and Fission in Small Groups. **1977**. *Journal of Antropolological Research* **33**, 452–473.

[11] Zhang, Y., Friend, A.J., Porter, M.A., Fowler, J.H., and Mucha, P. J. In preparation.