Paper One

Modularity and community structure in networks

Motivation

Community detection is widely used in many fields. However, how to evaluate the result of community detection remains a problem. The most effective way that has been introduced is modularity. Unfortunately, the approaches that try to maximize modularity are computationally expensive. Therefore, the authors try to figure out a new method that can work more efficiently and get a good result as well.

Related Theory

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j,$$

- Modularity: , thus it can be rewritten as $Q = \frac{1}{4m} s^T B s$, where $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$ named modularity matrix.
- Modularity matrix: $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$. Since the elements of each of its rows and columns sum to zero, it is similar to Laplacian matrix and has many similar features as well.
- Generalized modularity matrix: $B_{ij}^{(g)} = B_{ij} - \delta_{ij} \sum_{k \in g} g_{ik}$. It reduces to modularity matrix when the network is complete.

From the above mathematical theories, we can know that if we want to

maximize the modularity, then we can choose some largest eigenvectors to divide the responding graph, which is also similar to the idea of Spectral Clustering.

Main algorithm

Two communities:

1. Compute the adjacency matrix and the degrees of each vertex
2. Construct the modularity matrix
3. Compute the eigenvalues and eigenvectors of the modularity matrix
4. Group the vertices according to their signs of the elements in leading eigenvector

   (Typically, if the modularity matrix has no positive eigenvalues, the network is indivisible.)

More than two communities: (Hierarchical clustering)

1. Construct the modularity matrix $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$
2. Compute its leading eigenvalue and leading eigenvector
3. Divide the network into two parts according to the signs in leading eigenvector
4. While not converge do

   Construct the generalized modularity matrix $B_{ij}^{(g)} = B_{ij} - \delta_{ij} \sum_{k \in g} g_{ik}$ for each part

Do as the step 2 ,3 and 4 in each part

The convergence condition is that there is no positive contribution to the total modularity.

My idea

The algorithm is not computationally expensive, and it can also automatically determine whether the community needs to be divided or not. Its idea is very similar to Spectral Clustering, which aims to minimize the Ncut while the algorithm introduced in this paper aims to maximize the modularity. The algorithm is not hard to implement since it can be implemented in breath-first strategy and we can use the STL's queue to achieve this goal.

Paper two

Semi-supervised Community Detection Framework Based on Non-negative Factorization Using Individual Labels

Motivation

In the past, the algorithms used in community detection had their focus on topology information but ignored the priori information that we could get in advance. Therefore, the authors try to utilize the priori information to improve the performance of community detection. Moreover, authors are curious about the effect of so-called important nodes thus having an experiment on them.

Related Theories

- NMF: Use UV matrix factorization to present original adjacency matrix. Its idea is factorize matrix A $\in R^{N\times N}$ into two matrix U $\in R^{N\times K}$ and V $\in R^{N\times K}$, where K is the target number of communities. Thus, the objective function of NMF is to minimize the equation: $L_{LSE}(A,UV^T)=\left\|A-UV^T\right\|_F^2$. And the iterative updating formula is as followed:

$$U_{ij}\leftarrow U_{ij}\frac{(AV)_{ij}}{(UV^TV)_{ij}} \quad \text{and} \quad V_{ij}\leftarrow V_{ij}\frac{(A^TU)_{ij}}{(VU^TU)_{ij}}$$

- Encoding the prior information: If a node does belong to a cluster, it has PL(positive label). If a node does not belong to a cluster, it has NL(negative label). And the prior information matrix O can be defined with the idea of the theory of possibility as followed:

If node i has PL, and i belongs to j-th community,

$$O_{ik}=\begin{cases}1, & k=j\\ 0, & k\neq j\end{cases} \quad for\ k=1,2,...K$$

If node i has NL, and i cannot belong to j-th community,

$$O_{ik}=\begin{cases}0, & k=j\\ \dfrac{1}{K-1}, & k\neq j\end{cases} \quad for\ k=1,2,...K$$

If node i has no priors,

$$O_{ik}=\frac{1}{K} \quad for\ k=1,2,...K$$

Main algorithm

The general idea: From NMF, a node i will belong community k if V_{ik} is the

largest among the i-th row of V. Therefore, we can embed O into V with dot product: V = O$\times$V, from which we can enhance the value of V_{ik} if the node i belongs to community k and reduce the value of V_{ik} if the node i does not belong to community k.

Moreover, this paper introduces a penalty term to the original objective function: $\min_{U\geq 0,V\geq 0}\left\|A-UV^T\right\|^2+\lambda\sum_{k=1}^{K}\left\|V_k\right\|_1$, thus its updating formulas change as well：

$$U_{ij}\leftarrow U_{ij}\frac{(AV)_{ij}}{(UV^TV)_{ij}}$$ （unchanged）

$$V_{ij}\leftarrow V_{ij}\frac{(A^TU)_{ij}}{(VU^TU)_{ij}+\lambda}$$

The Algorithm：

1. Set the parameter λ
2. Construct matrix O
3. Initialize U,V randomly but they must be positive matrices
4. While not converge do

   Set $U_{ij}\leftarrow U_{ij}\frac{(AV)_{ij}}{(UV^TV)_{ij}}$

   Set $V_{ij}\leftarrow V_{ij}\frac{(A^TU)_{ij}}{(VU^TU)_{ij}+\lambda}$

   Normalize U, V

Other Work in this paper

In the research field, many methods consider some nodes more important

than other nodes according to their degrees, betweenness, shortest paths, the weights of nodes and edges and so on. Here are definitions of degree and betweenness:

- Degree: the number of edges connected to the node in the network, whose formula may be: the degree of node i $d_i = \sum_{j=1}^{N} a_{ij}$, where a_{ij} is the element in the adjacency matrix A.
- Betweenness: the betweenness of node u refers to all the shortest paths through node u and its normalization formula is $B_u = \sum_{s \neq t \neq u} \frac{\sigma_{st}(u)}{\sigma_{st}}$, where σ_{st} presents the number of shortest paths from node s to node t, and $\sigma_{st}(u)$ presents the number of shortest paths from node s through node u to node t.

However, from the experiments in this paper, authors show us that these so-called important nodes (defined by degree or betweenness) do not have such effects as we thought before.