Evaluation for

Community-finding Algorithms

Instructor: John Hopcroft

Scribes: Xingkai Wang

Fangkui Zhang

Zhipeng Chen

Shanghai Jiao Tong University

2013-1-18

【Introduction】

This is a report on evaluation of the algorithm for detecting community structure. We will describe how we can evaluate an algorithm for finding community structure in this report. For the purpose of theory rather than applications, we only evaluate its accuracy rather than efficiency. We will use the algorithm below to judge whether an algorithm is good enough.

【Part 1】Method to evaluate algorithms

It is a fact that the answers which community-finding algorithms give us may not be exactly correct. We define some variables in order to evaluate how good it is for the outputs.

Firstly, we define the differences between two community structures. Suppose we have community structures A and B(A and B are graphs). For every community C1 in A(C1, C2 are sets of vertices), Find a community C2 in B such that |C1C2| ≥ . The value of difference is v = |C1| - |C1 C2|. If we failed to find C2, then halt. And it means the two structures are too different. If some are found, choose one randomly. For every community C1 in B, in the same way, we calculate all the differences v. In the end, we make a summation of all v and get V = .

Then we calculate V between the resulting community structure our algorithm gives us and the real data. V can evaluate how good two sets of communities fit. That is how good a community-finding algorithm runs. It is obviously correct that the answer is better and better as V gets smaller and smaller.

There is a fact that when we consider C1 we may find C2 attached to it, while when we consider C2 we may not get C1 symmetrically, just because there are a lot of communities can be attached to C2 and we just choose one of them randomly. In some sense, we do not care the appearance because it cannot change the total solution obviously.

【Part 2】Creating random graph

After a group discussion, we decided to design some random graph in order to test the accuracy of those community-finding algorithms.

Firstly, we decided to generate a graph with communities already created. That is to say, we generate vertices first. And put some of vertices in a community manually, and some other vertices in another community, and so on. Then we got a graph with a lot of vertices separated in several communities. Secondly, we generate edges between two vertices in order to let communities are more likely to be real communities. That means, we create much more edges inside each community than between communities. As a result, we have generated a graph which can be used as a test to evaluate algorithms.

It is not such an easy problem as we thought of forehead to generate such a graph, because we need an algorithm for adding edges to our graph with only vertices in it. We tried to fix the degree of each vertex on the first attempt. We wanted to control the total number of the edges which is connected with vertices inside the same community. And so does the number outside. However, we just cannot find such an efficient algorithm to complete it. So we decided to control the probability inside and outside in order to create the graph, with a much bigger probability inside than outside. That is a much easier job for us to create a random graph.

Finally, we changed the total number of vertices n, number of communities c, probability inside Pin and probability outside Pout, and we got a lot of random graphs by the same mean.

【Part 3】Evaluating local-partition algorithm.

In this part, we will use the random graphs we created by means of part two and the formula we created in part one to evaluate algorithms.

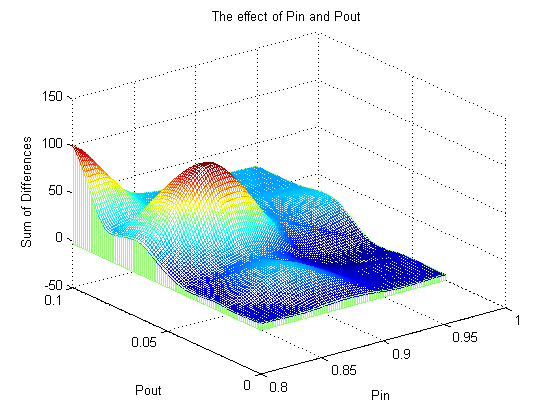
Mr. Shang is also researching community structure. He is interested in finding new algorithms. And we use our data to evaluate his local-partition algorithm and the result shows as follows.

We can conclude that the total number of vertices in a graph do not affect the accuracy of the algorithm via testing a large amount of data. It is the total number of communities, the relationship between pin and pout that affect the accuracy obviously.

We can get the chart following by our test. The percentage means the proportion of the good answers in all answers. We say an answer is good in case that the V is at a low degree.

It means that as the number of communities grows, the accuracy goes down sharply. When the number came to 5 or larger, the algorithm cannot find communities in the graph correctly.

We let the total number of communities be 4. We make some tiny changes on pin and pout. And we got the answer as follows:



We can conclude that the answer is better with a bigger Pin and a smaller Pout. We can also find that the algorithm do not have a good answer in some random graph with certain Pins and Pouts. This may be led by the uncertainty of the algorithm.

【Part 4】Reference

[1] M. Girvan and M. E. J. Newman, Community structure in social and biological networks. (2001)

【Part 5】Acknowledge

We appreciate the great work of Professor John Hopcroft. We learned a lot from communicating with him in the courses. We also reference the slide and presentation provided by Jingbo Shang. Thanks to all the people who helped us finish our work.