Examples: Community Mining

The Girvan-Newman algorithm is similar to a divisive hierarchical clustering algorithm, as we discussed earlier. It starts from the full network. The algorithm then uses the betweenness to identify nodes that sit between communities. Remember that the betweenness counts the number of times that a node or edge occurs in the geodesics of the network. The algorithm proceeds as follows:

In step 1, the betweenness of all existing edges in the network is calculated first.

In step 2, the edge with the highest betweenness is removed.

In step 3, the betweenness of all edges affected by the removal is recalculated.

Steps 2 and 3 are then repeated until no edges remain and every node corresponds to its own community. Just as with clustering, the result of the algorithm is essentially a dendrogram.

Let's illustrate this with an example. Here you can see an example network. Which edge do you think has the highest betweenness? The edge between G and H has the highest betweenness. What's the actual value for the betweenness? The value for the betweenness is 49, because there are seven nodes in the left community and seven nodes in the right community and 7 times 7 = 49. So, we remove the edge between nodes G and H. This creates two communities. We can now continue the analysis. The next edges to be removed are the edges between nodes C and G, between nodes E and G, between nodes H and I, and between nodes H and L. Here you can see the result of this. In a next step, we can now remove all remaining edges. This bring us to the nodes with no more edges.

To summarize, here you can see a visualization of the hierarchical network decomposition. Similar to what was discussed before, in a clustering context, a key decision is how to determine the optimal number of communities. In other words, how can we measure the quality of a particular set of communities or the modularity?

Suppose we have k communities. We now define a k-by-k symmetric matrix E with entries eij specifying the fraction of all edges in the network that link nodes in community i to nodes in community j. The trace of this matrix is the sum of the diagonal elements, or Trace (E) equals the sum across ii of eii, which gives the fraction of edges in the network that connect nodes in the same community.

A good division into communities or modularity should have a high value for the trace. However, if all nodes were put in their own communities, then a maximal value of the trace equal to 1 would be obtained. Clearly, this is not desirable. Hence, let's define the row or column sum ai as follows: ai equals the sum across j of eij. This represents the fraction of the edges of the network connecting to nodes in community i. If the communities are randomly connected, we have eij equals ai tims aj.

The Q-modularity can now be defined as follows: the sum across i of eii minus ai squared. Hence, it measures the fraction of within-community edges in the network, which is Trace (E), minus the fraction of within-community edges in a network that has the same communities, but with random connections between the nodes. In the case of random communities, Q will equal zero. For strong communities, Q will approach 1. In practice, Q values between 0.3 and 0.7 indicate a significant community structure.

The Q measure can then be monitored as community mining proceeds and the optimal community solution can be found where it reaches a peak. You can see this visualized here. To the left is the dendrogram that is produced by the Girvan-Newman community-mining algorithm. To the right is the Q-modularity graph. The red dashed line indicates the peak and thus the optimal number of communities.

In this case, the optimal number of communities is three. Community 1 consists of nodes A, B, and C; community 2 consists of nodes D, E, and F; and community 3 consists of node G.

Social Network Analytics

Close