Rakesh Kumar Vakkalagadda

Papers – "Detecting Community Structure in Networks"
- M.E.J. Newman

"Distributed Community Detection in Delay Tolerant Networks"
- Pan Hui, Eiko Yoneki, Shu-Yan Chan, Jon Crowcroft

Questions from the presentation

In Kernighan-Lin algorithm, nodes are assigned edges on what basis? random? why?

The input to the Kernighan- Lin algorithm is a graph that represents a network where edges are assigned based on the encounters with other nodes. The algorithm also requires the user to specify the size of the two groups into which the network should be split and to choose a starting configuration for the groups, for example by dividing the vertices at random. Then, we consider all possible pairs of vertices in which one vertex is chosen from each of the groups, and calculate the change 'Q'(in the benefit function – is the number of edges that lie within the groups minus the number of edges that lie between them) that would result from swapping them. Then we choose the pair that maximizes this change and perform the swap.

What is good, bad ugly about the Kernighan-Lin algorithm?

Good – The good thing about this algorithm is that it assigns something called a benefit function to divisions of the network and then attempts to optimize that benefit over possible divisions. The benefit function is the number of edges that lie within the two groups minus the number that lie between them. Hence, it checks all the possible combinations and divides in the optimal way.

Bad – The disadvantage with this approach is that we need to specify the sizes of the communities before the algorithm starts.

Ugly – The above disadvantage makes the algorithm unsuitable for most applications to real-world network data in which we have no knowledge of the group sizes. The idea that we can run the algorithm for a variety of group sizes is also not the answer because in addition to increasing the run time of the algorithm to $O(n^{3)}$),it may not work since the best values of Q are obtained for a very asymmetric divisions of the network. And moreover, the algorithm only divides the network into two divisions and the divisions into further groups is obtained by repeated bisection but the problem is that we do not clearly know when to terminate this process.

Complexity of the algorithm?

This algorithm has a worst case complexity of $O(n^2)$.We consider all possible pairs of vertices in which one vertex is chosen from each of the groups, and calculate the change in 'Q' (benefit function) that would result from swapping them. Then we choose the pair that maximizes this change and perform the swap. When all vertices in one of the groups have been swapped once, we go back over the sequence of swaps that were made and find the

point during this sequence at which Q was highest. This is taken to be the bisection of the graph.

What is Modularity ?

Modularity is a measure how good a particular division is. For a division with g groups, we define a g *g matrix e whose component e_{ij} is the fraction of edges in the original network that connect vertices in group i to those in group j. Modularity is defined as:

$$Q= \Sigma e_{ii} - \Sigma e_{ij}e_{ki} = Tr\ e - ||e^2||$$

where || x || indicates the sum of all elements of x. 'Q' is the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random without regard for the communities. A value of Q = 0 indicates that the community structure is no stronger than would be expected by random chance and values other than zero represent deviations from randomness. Local peaks in the modularity during the progress of the community structure algorithm indicate particularly good divisions of the network.

Why use distributed algorithm? What is the need?

Many centralized community detection methods have been proposed. These centralized methods are useful for offline data analysis on mobility traces collected to explore structures in the data and hence design useful forwarding strategies, security measures, and killer applications. But as self-organizing networks, we would also ask whether the mobile devices can sense and detect their own local communities instead of relying on a centralized server which leads to the distributed community detection. These kind of distributed community detection helps in the mobile devices to identify there own communities and forward the packets to its neighbors based on its own view of the network and the neighbors view which it would transfer the packet too.

Complexity for distributed algorithms? How much time it takes?

I have few doubts regarding this and am completely not clear with the complexities. I have mailed the authors as I had some queries. I will answer this as soon as I get a reply from them.

Why did you pick these three?

All the algorithms discussed were discussed in the paper "Detecting community structure in networks" by M.E.J. Neumann. I felt the author presented a very good case study of algorithms starting to identify groups based on the computer science perspective and sociological perspective. As identifying groups involves both these dimensions, I felt we could apply the above algorithms in the traces we collected and would be a good starting point.

What is the difference from other clustering algorithms? There are other clustering algorithms? Did we read other algorithms?

I haven't really read gone through other clustering algorithms prior to this. I felt that this algorithms presented in the paper gave an overview about the possible dimensions one could look at while detecting communities from the traces. I have checked and found a lot of clustering algorithms used in machine learning, data mining, pattern recognition etc. such as k-means, fuzzy c-means, QT etc. I have gone through these algorithms but the algorithms which I presented would be more relevant to apply in our project as they divide into communities based on some similarity measures like in the case of hierarchical clustering which is the second algorithm I discussed two nodes are said to belong to the same community and are connected by an edge if it has the same set of neighbors and this is a reasonable intuition because people belonging to the same community are likely to meet regularly. The edges added in this way have no direct connection with the edges of the original network whereas in the algorithms which I have gone through till now the clusters are decided based on the original network.