Due midnight Friday, November 12 2010

CS224W: Social and Information Network Analysis

Competition - Fall 2010

This competition must be done completely individually, but it is not mandatory. However, the five students with the highest scores will get extra credit: 10% for first place, 8% for the second place, 6% for the third, and 4% for fourth and fifth. In the case of ties, multiple students may receive the extra credit amount listed above for the given place.

Clustering of Signed Graphs

In this task your goal is to identify communities (coalitions) in a network with positive and negative edges. Note that you are not allowed to use any community detection software, but can use packages for general network analysis like Networkx, JUNG, and SNAP.

Networks with signed edges were first addressed by structural balance theory. The structural balance considers the possible ways in which triangles on three individuals can be signed, and posits that triangles with three positive signs (three mutual friends) and those with one positive sign (two friends with a common enemy) are more plausible — and hence should be more prevalent in real networks — than triangles with two positive signs (two enemies with a common friend) or no positive signs (three mutual enemies). See Figure for an illustration. Balanced triangles with three positive edges exemplify the well-known principle that "the friend of my friend is my friend," whereas those with one positive and two negative edges capture the notions that "the friend of my enemy is my enemy," "the enemy of my friend is my enemy," and "the enemy of my enemy is my friend." The network is said to be balanced, if all triads in a network are balanced.

The question whether a balanced network can be divided into separate parts arises naturally. The challenge is to define coalitions of nodes such that there are only positive links within coalitions and negative links are between them. Cartwright and Harary showed [1] that if a complete network is balanced, it can be split into two opposing coalitions (and vice versa). It is easy to see how this can be done: First, pick a random node to initialize one of the coalitions, and put it all its "friends" on one side and all its "enemies" on the other. If there were a negative edge between two of the friends, or if there were a positive one between a friend and an enemy, the network would be unbalanced. Since the network is a clique, all the nodes have been classified into one of the coalitions.

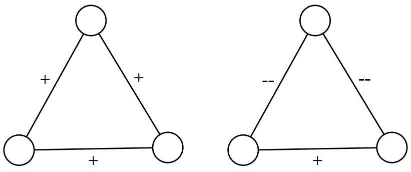

Figure 1: Balanced triads

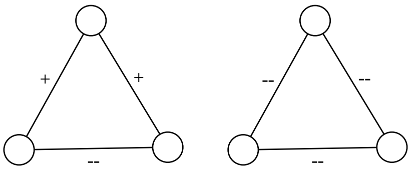

Figure 2: Unbalanced triads

In reality, however, social networks are rarely perfectly balanced. The question then becomes whether we can still cluster nodes in some meaningful way. Obviously, there are some links that make a network unbalanced. The number of such links can be expressed as an amount of *frustration*. Links that contribute to *frustration* are negative links within coalitions and positive links between coalitions. So, in this task you will try to partition a graph into two coalitions such that the *frustration* is minimized. In a reality there may be multiple coalitions, but we would still get the same amount of frustration.

So given a network with positive and negative signs your task is to split the nodes of the network into two partitions such that the number of positive edges between the members of the partition and the number of negative signs between the partitions is maximized. For every "correct" edge you get 1 point and for every edge that is "incorrect" (i.e., $-$ edge inside the partition or $+$ edge between the partition) you get -1 point.

Formally, let G be a graph with n nodes. We define the entries of the adjacency matrix $A \in \mathbf{R}^{n \times n}$ of G as follows: if a positive link is present from node i to node j, $A_{ij} = 1$, if a negative link is present, $A_{ij} = -1$, and $A_{ij} = 0$ otherwise. We separate the negative and positive links by setting $A^+_{ij} = A_{ij}$ if $A_{ij} > 0$ and zero otherwise, and $A^-_{ij} = -A_{ij}$ if $A_{ij} < 0$ and zero otherwise, so $A = A^+ - A^-$.

We want to assign each node i to either community σ_1 or community σ_2, where $\sigma_1, \sigma_2 \subseteq \{1, \ldots, n\}$ while minimizing:

$$f(\sigma_1, \sigma_2) = \sum_{ij} A^-_{ij} \delta_{ij}(\sigma_1, \sigma_2) + A^+_{ij}(1 - \delta_{ij}(\sigma_1, \sigma_2))$$

where

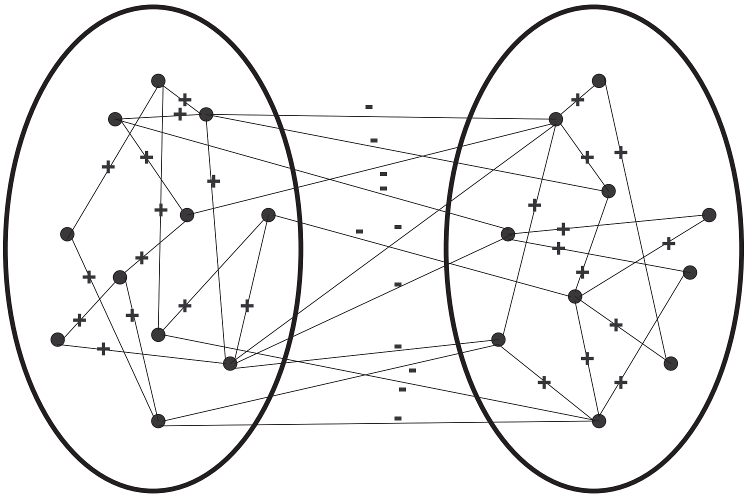

Figure 3: Balanced networks can be split into two coalitions with positive ties within the members of the coalition and negative ties between the coalitions.

$$\delta_{ij}(\sigma_1, \sigma_2) = \begin{cases} 1, & \text{if } i, j \in \sigma_1 \text{ or } i, j \in \sigma_2 \\ 0, & \text{otherwise.} \end{cases}$$

This minimization problem is NP hard, so we do not expect you to find a general method for solving it. However, there are several interesting heuristics that provide reasonably good results. Your job is to find your own heuristic and report its performance on the given data.

Some heuristics to get you started

The following are some simple heuristics you could try. Feel free to use them to get a starting point or to refine the result of your heuristic.

- Greedy: Split the nodes randomly into two groups. Then go through the nodes one by one finding out whether switching sides would improve the value of the objective. Keep doing this until no improvement can be gained by a node switching the side.
- Clustering: Run one of the clustering algorithms mentioned in class (for example k-means or Min-Cut) taking into account only the positive edges. That should give you two clusters with lots of positive edges within them.
- Max-cut: If we focus only on the negative edges instead, the problem becomes a max-cut. The max-cut problem is also NP-hard, but you might be able to use some approximation algorithms or heuristics.

Data

You will be partitioning a graph derived from the results of an election of Wikipedia administrators. (A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance.) You may think of this graph as directed (i.e. it may contain an edge going from node a to node b, as well as one going from node b to node a, and these two edges are distinct). The graph contains 7,115 nodes, and 103,689 edges. We will evaluate the cost function above on all edges in the graph.

Each line in the text file specifies the following information:

```
<node_1>  <node_2>  <the sign of the edge>
```

Please download this data from the course website.

Submission

You should turn in a paragraph explaining your method. You should also turn in a separate .txt file where each line consists of a node and the partition it belongs to, with line breaks between lines:

```
<node_id>  <partition_id>
<node_id>  <partition_id>
.
.
.
```

Please upload both files to your dropbox folder, clearly labeled “yourName_competition_writeup.txt” and “yourName_competition_solution.txt”.

References

[1] D. Cartwright and F. Harary, “Structural balance: a generalization of Heider’s theory”, *Psychological Review*, (1956), 277-293.