Algorithm of point of view driven community detection and visualization of socio-semantic social networks

Juan David Cruz  
Dpt. LUSSI, Telecom-Bretagne  
juan.cruzgomez@telecom-bretagne.eu

Cécile Bothorel  
Dpt. LUSSI, Telecom-Bretagne  
cecile.bothorel@telecom-bretagne.eu

François Poulet  
University of Rennes I – IRISA  
francois.poulet@irisa.fr

Abstract—Information contained in social networks is more than just connections among actors. Each actor has self information which can be used to analyze the network from different perspectives. In this paper we present a community detection and visualization model designed to use both types of information and identify non-evident information from they.

I. INTRODUCTION

Explosion in the use of social networking sites has created a space in which the information about people and their connections is increasing all the time. The amount of information allows to do different kind of analyses to discover information beyond the evident. Today, most techniques focus on one or another kind of information, and in the case of visualization, methods try to show the difference between groups instead of their interactions.

In this paper we present a method to detect and visualize communities in social networks using both, the non-structural and the structural information of the actors in order to identify key actors and to analyze the composition of groups and analyze their interactions.

The paper is organized as follows: in Section II a brief review of previous relevant work is presented, in Section III the problem is stated and the general proposed approach is presented, then in Section IV conclusion and future work is presented.

II. RELATED WORK

A. Community detection

The basic idea of community detection is to find groups of densely connected nodes. To do that, several measures have been proposed such as those presented in [1] and [2]. However, one of the most used measures [3] is the modularity $Q$, proposed by Newman [4], compares the fraction of the edges within each cluster with the fraction of edges among clusters, i.e., the intracluster edges density versus the inter-cluster sparsity. The modularity compares the fraction of the edges within each cluster with the fraction of edges among clusters, thus, a higher modularity means that the proportion of the edges falling into clusters are greater than the edges between them. However, the direct calculation of the modularity is an $O(n^3)$ operation.

To reduce the calculation time of the modularity, [5], have proposed a greedy agglomerative algorithm to find communities. In the first step each node is assigned to one community and the initial modularity is calculated. Then, each node $i$ is removed from its community and moved iteratively to each community. After each movement the modularity gain is calculated, and $i$ will be assigned to the community giving the largest positive gain of $Q$. If no positive gain is possible, $i$ remains in its original community. This process is applied iteratively until no further improvement can be achieved and no individual move will improve the modularity. This algorithm is executed in linear time for sparse graphs [5].

The previous methods only consider topological properties of the Social Network. The closest related work is the one by [6], who present a community detection approach using structural and attribute similarities. They use a random walk throughout a predefined set of $k$ clusters, and tries to maximize the distance between clusters by moving nodes according to their similarity. First, they create an augmented graph from the node attributes, then they execute the random walk over the transition matrix generated by the augmented graph. This leads to find $k$ groups of semantically close nodes. To measure the clustering from a structural point of view, they use the density of edges within the clusters.

B. Community visualization

A clustered graph can be defined as a particular type of hierarchical graphs in which the distance $\lambda$ between two nodes from different levels in the inclusion tree is one at maximum. This tree can be seen as the dendrogram derived from a hierarchical clustering process.

From this definition, Eades et al., [7] propose a layout method to exploit this structure in a recursive way using a force directed algorithm and a layered 3D visualization. First, the nodes within each community are placed in the first layer of the 3D representation using a force directed algorithm. Then another layer is added above the first one. In this layer a circle representing each community is drawn in the centroid of the nodes placed in the previous step. This operation is repeated throughout all levels in the inclusion tree.

Tamassia [8] presents an algorithm which uses rectangles to represent nodes and straight lines with right angles to represent links. This kind of layout is used also by di Giacomo [9] to present a synthetic view of Web graphs. In this case the rectangles are big enough to contain the nodes from each community, which can be problematic with huge graphs.

Noack [10] presents a force directed algorithm which uses at the same time a linear model to represent the node attraction between adjacent nodes and a logarithmic model for the repulsion between non connected nodes. This approach does not use a clustered graph but it uses the LinLog model to find the partition while the graph is being drawn.

III. PROBLEM DEFINITION AND PROPOSED APPROACH

As presented in Section II, most approaches to community detection are based only on the structure of the social graph, an on the other hand, visualization algorithms have been designed to show the boundaries of groups and not to show the relationships between them.

In this work we want to explore different approaches to find a model which allows us to answer two main questions:

1) How to use the semantic information to influence the community detection process, and,

2) How to visually identify interactions and key actors from a graph partition.
Figure 1 shows the general architecture of the designed model. This model uses an augmented network, which is a social graph with the additional semantic information of its actors, and finds a semantic-aware partition. Using this partition, a visualization algorithm is used to reveal the interactions occurring between communities.

A. Community detection step

This step is performed in tow stages: first, using the semantic information of the augmented network, a clustering is performed to find groups of nodes semantically close. Using the intermediary partition $C_F$, the weights of the edges of the social graph are changed. This change is made to reflect into the graph structure the proximity of connected nodes from a semantic perspective, i.e., edges connecting nodes in the same semantic group will have a stronger connection.

Thus, the first step is to semantically group the nodes, i.e., according to their non-structural information contained in the augmented social network. This step is performed using an unsupervised clustering algorithm: Self-Organizing Maps [11], which defines a lattice of neurons, each one with a weight vector of the same dimensionality of the elements in the data set. Each element is presented to the map and one neuron, the closest, is marked as winner. This process is performed with the whole data set until the network converges.

The outcome of SOM is a partition of the nodes according to their non-structural information. Using this, each edge of the graph is evaluated to determine if both ends belong to the same semantic group, if it is the case, the edge is strengthened by increasing its weight. This change will influence the modularity based community detection algorithm by privileging communities with stronger edges.

B. Community visualization step

Once the partition is produced, we proceed to layout it. The goal of the layout algorithm is to show the interaction between communities and hence discover important nodes. To do this, we take the set of nodes and divide it into two groups: one with nodes with neighbors in other communities, called border nodes, and other with nodes connected with nodes in their own community, called inner nodes.

Thus, the algorithm place the border nodes in the interaction zone using multi-dimensional scaling – MDS: this allows to place nodes according to their structural similarity, i.e., nodes with similar neighbors will be placed close. After the border nodes are placed, inner nodes from each community are located in a ring outside the interaction zone.

Figure 2(a) shows the placement of a clustered network and its communities. Nodes at the center are the border nodes, the nodes outside are the inner nodes from each community. Figure 2(b) shows the schema of placement for each element of the drawing. All the coordinates of the nodes are normalized to be in the interval $[0, 1]$.

This representation allows to find at the interaction zone spanner nodes, broker nodes and nodes which in general are central for the communication of communities. Some roles can be found in [12].

IV. CONCLUSION AND FUTURE WORK

In this brief paper we have presented a model for finding and visualizing communities in social networks. In the first case, the communities found by our algorithm contain similar nodes in terms of the non-structural information they have and each group is well connected. During the experiments we noticed a natural trade-off between the quality of the semantic partition and the structural partition: increase one will affect the other.

In the case of the visualization method we propose an approach which allows to visualize and analyze the interactions between communities. This is made by focusing on the nodes, in each community, which effectively share connections with other groups.

Future work includes the addition of exploration and navigation features to help the analysis of the clustered visualization.

REFERENCES