Visualization of the Interactions Between Communities in a Social Network

J. Cruz$^1$  C. Bothorel$^1$  F. Poulet$^2$

$^1$LUSSI Department
Telecom – Bretagne

$^2$IRISA
Université de Rennes 1
1. Introduction
   - Motivation
   - Challenges

2. Previous Work

3. Our Approach
   - Previous Elements
   - The algorithm

4. Conclusions and Perspectives
A graph partition $C_G$ represents the different clusters, or communities, obtained from the graph $G$. These communities interact with each other and have different configurations.

**Given a partition $C_G$, how to visualize it?**

This visualization should help to visualize the interaction between communities and their composition.
Communities Graph
Layout
Community Graphs?

What is it?

- Communities Graph Layout
- Clustered Graph
- Social Graph
- Inclusion Tree

Introduction
Previous Work
Our Approach
Conclusions and Perspectives
Appendix

CRUZ, BOTHOREL, POULET
Community Interaction Visualization
Community Graphs?

What is it?
Identify and visualize the interactions between communities in the SN

Main goal?

Clustersed Graph

Social Graph

Inclusion Tree

Communities Graph Layout

Introduction
Previous Work
Our Approach
Conclusions and Perspectives
Appendix

CRUZ, BOTHOREL, POULET
Community Interaction Visualization
Community visualization is a specific case of graph layout; it has in general the same challenges:

- Algorithm Complexity
- Graph Readability
- Display Clutter
- Graph Navigation

Additionally, in our work we want to:

- See the interaction between communities
- Navigate throughout points of view (projected)
Previous Work

Approaches

■ Multilevel visualizations
  (Eades [1])

■ Clustered weighted graphs
  (Auber et al., [2])

■ Layout of overlapping
  clustered graphs
  (Santamaria [3])

■ Orthogonal rectangular
drawings
Approaches

- Multilevel visualizations (Eades [1])
- Clustered weighted graphs (Auber et al., [2])
- Layout of overlapping clustered graphs (Santamaria [3])
- Orthogonal rectangular drawings
Previous Work

Approaches

- Multilevel visualizations (Eades [1])
- Clustered weighted graphs (Auber et al., [2])
- Layout of overlapping clustered graphs (Santamaria [3])
- Orthogonal rectangular drawings
**Approaches**

- **Multilevel visualizations** (Eades [1])
- **Clustered weighted graphs** (Auber et al., [2])
- **Layout of overlapping clustered graphs** (Santamaria [3])
- **Orthogonal rectangular drawings**
Quotient Graph $\mathcal{G}$

The graph representation of the graph clusters: each node of $\mathcal{G}$ contains the nodes of $G$. The edges of $\mathcal{G}$ represent the intercluster edges.
Inner Nodes $V_C$

These are the nodes with edges from/to nodes in the same cluster only.
Border Nodes $V_C$

These are the nodes connecting communities: have neighbors in other clusters.
Multi–dimensional Scaling

Is a technique for representing dissimilarities among objects in some representation and map them as distances into some $\rho$-dimensional space.

- Needs a dissimilarity measure $\delta$ to be mapped into distances.
- Finds a set of coordinates by iteratively reduce the error (stress) $\sigma$ between the distances and the dissimilarities. Typically: $\sigma = \sum_{i<j} (d_{ij} - \delta_{ij})^2$.
- There are three variants of MDS:
  - Analytical solutions
  - **Stress reduction (gradient descent)**
  - Spring–mass simulation stress function
Nodes Dissimilarity Measures

Jaccard distance $\delta_J$

Given two nodes $a$ and $b$, the Jaccard (dis)similarity measure will calculate the proportion of common neighbors of $a$ and $b$:

$$\delta_J = 1 - \frac{|N_a \cap N_b|}{|N_a| \cup |N_b|}$$

Thus, $a$ and $b$ are similar if the have a similar neighborhood.

We use the Jaccard distance because it describes also the variability of the neighborhood.

Shortest Path

Given two nodes $a$ and $b$, the distance between them is given by the shortest path $SP_{a,b}$. This yields a matrix

$$\Delta_{SP} = SP_{i,j}, \forall i, j \in V$$

$V$ is the set of nodes.
The quotient graph is the representation of the groups as nodes.

- All happens in a circumference of radius 1
- The border nodes will be placed in the inner circle ($r = 0.5$)
- The inner nodes will be placed in the ring ((0.5, 1.0])
- The center of each cluster will be placed on the circumference of radius 0.75
The border nodes represent the connections between the communities in the graph.

- In this phase, the first step is to create a dissimilarity matrix $\Delta \overline{V_C}$ from all the border nodes in the graph.
- Then, the MDS algorithm is executed for this set of nodes using $\Delta \overline{V_C}$.

The result of the algorithm are the coordinates for each node and the location of each community.
Each cluster is individually handled, executing MDS for each one.

- For each community $i \in V_C$ a dissimilarity matrix $\Delta_{V_{C_i}}$ is calculated.
- The MDS algorithm is executed for each community using their $\Delta_{V_{C_i}}$ matrix.
- The coordinate of each node is transformed to be placed around the center of its community.
In general, the complexity of MDS, for a graph of $n$ nodes is $T(n) = O(n^3)$

- Dividing the problem according to the communities number reduces the expected complexity and allows to manipulate each community easier.

- In general: $\hat{T}(n) = O\left(k + \frac{(n-|V_C|)^3}{k^2} + V_C^3\right) < T(n)$

- The worst case: all nodes are border nodes. $\hat{T}(n) = T(n)$

The complexity can be reduced implementing the matrix multiplication algorithms using block matrices and parallelizing the process.
Network of intra-school links from Facebook 100 subset [4] extracted in 2005
- Containing 6386 nodes and 435324 edges
- 12 communities, 5973 border nodes
- Communities converge near the origin: zone of higher interaction between communities
Some Results 2

- Graph extracted from a Twitter dataset
- 5389 nodes and 27347 edges.
- 15 communities, 1385 border nodes
- The border nodes show a different interaction pattern: even a community with a low interactivity pattern.
We have developed a MDS based algorithm for clustered graphs that divides the nodes according to the type of the edges.

This allows to see how the different communities interact identifying “bridge” nodes between communities and interaction patterns.

This division produces also a complexity reduction due to division of the dissimilarity matrix into $k + 1$ matrices.
Perspectives

- Improve the visualization results by: reducing edge crossings and node cluttering
- Work with an interactive model for the visualization
- Improve the implementation to take advantage of parallel architectures e.g., execute in parallel layout of the inner nodes
- Analyze, from a social point of view, the results of the layout and the way how the communities interact
