Semantic Clustering of Social Networks using Points of View

Juan David Cruz* — Cécile Bothorel* — François Poulet**

*Département LUSSI, Télécom – Bretagne
{juan.cruzgomez, cecile.bothorel}@telecom-bretagne.eu
**Université de Rennes 1 – IRISA
francois.poulet@irisa.fr

RÉSUMÉ. Les algorithmes classiques de détection de communautés dans les réseaux sociaux utilisent l’information structurelle pour détecter des groupes, i.e la topologie du graphe de relations. Toutefois, ils ne prennent en compte aucune information externe qui peut guider le processus et aider à la réalisation des analyses du réseau selon différentes perspectives. La méthode proposée utilise de façon conjointe, l’information sémantique du réseau social, représentée par des points de vue, et son information structurelle. Elle permet la combinaison entre les relations sociales explicites, les arêtes du graphe social, et les relations implicites, dites sémantiques, correspondant par exemple à des intérêts ou des usages similaires.

ABSTRACT. Classic algorithms for community detection in social networks use the structural information to identify groups in the social network, i.e., how clusters are formed according to the topology of the relationships. However, these methods don’t take into account any semantic information which could guide the clustering process, and which may add elements to do further analyses. The method we propose, uses in a conjoint way, the semantic information from the social network, represented by the points of view, and its structural information. This information integrates the relationships, expressed by the edges on one hand, and the implicit relations deduced from the semantic information on the other hand.

MOTS-CLÉS : Analyse de Réseaux Sociaux, Réseaux Socio-sémantiques, Détection de Communautés, Graph clustering, Self Organizing Maps

KEYWORDS: Social Network Analysis, Socio-semantic Networks, Communities Detection, Graph Clustering, Self Organizing Maps

1re soumission à CORIA, le 05–12–2010
1. Introduction

A social network is composed of a group of people linked according to different types of relationships.

Within a social network may coexist several kinds of relationships, like friendship, partnership, familial or from the work, among others. The members of a social network may be described also by their affiliation to an enterprise, to some community, common interests, and even by their usage of online contents.

The enhanced information which describes the relationships and the people within the network composes the “semantic” information of the social network.

By using this semantic information, the social network analysis could be performed from different perspectives, not only from the structural one, i.e., the topology of the network.

Hence, we propose a method which combines information from the network topology and from the actors, or the network’s semantic information. This semantic information can be divided into subsets of information, called points of view. Then, a point of view can be defined as an ensemble of features which represents a state of the network under a given perspective.

Thus, the inclusion of the semantic information will provide a guide to the clustering process, and by this way, change the partition configuration according to the perspective, or point of view, selected to perform the analyses.

The paper is organized as follows. In Section 2 is presented some previous work in community detection in social networks, in Section 3 we present the definition of the point of view of social networks; in Section 4 we present the proposed clustering method. In Section 5 some experiments and preliminaries results are presented before the conclusion and future work.

2. Related Work

Several methods have been developed to find clusters in a graph, or this is equivalent, find communities in a social network.

In general, those methods have been defined as optimization problems where the objective function is the maximization of some quality index. The indices measure the quality of a partition $C$ based on the number of edges within the cluster and the number of inter-cluster edges.

A general framework for quality indices and some classic graph clustering algorithms are presented below.
2.1. Clustering Quality Indexes

A quality index is a measure of the number of links between communities and the number of links within each community. The idea is to find a cluster configuration minimizing the number of inter–cluster links and maximizing the number of intra–cluster links. Thus, given a partition \( C \in A(G) \), where \( A(G) \) is the set of all possible clusterings, and a graph \( G \), Brandes et al., (Brandes et al., 2008) define a general function framework for measuring the quality of the partitions found by some algorithm. This framework is composed of two independent functions \( f \) and \( g \) that measure the intracluster density and the inter–cluster sparsity respectively. This index is shown below:

\[
\text{index}(C) = \frac{f(C) + g(C)}{N(G)}
\]

where \( N(G) \) is a normalization function set to \( \max \{ f(C') + g(C') : C' \in A(G) \} \).

Using this framework, (Gaetler, 2005) and (Brandes et al., 2008) define three quality indexes: the coverage, which measures the weight of all the intracluster edges compared to the weight of all edges within the graph; the conductance, which is based on the observation that if a cluster is well connected, then, a large number of edges have to be removed in order to bisect it, and the performance, which defines the quality of a given cluster based on the “correctness” of the classification of a pair of nodes. This correctness depends on whether two connected nodes belong to the same cluster, or two not connected nodes belong to different clusters. Additionally, another index, the modularity \( Q \), proposed by (Newman et al., 2004), 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. This index is the most commonly used in the different clustering methods as presented by (Fortunato, 2010).

2.2. Conventional Graph Clustering Methods

Several methods have been developed to find clusters in a graph. Those methods aim to find groups minimizing the number of edges between clusters and maximizing the number of edges within the clusters. These approaches can find better partitions when the adjacency matrix of the graph is sparse (Fortunato, 2010).

The algorithm proposed by (Newman, 2001) iteratively finds and removes the edge with the highest betweenness score. This process allows to find groups which are loosely connected between them and with well connected nodes within the group. The main drawback of this approach is the complexity of the calculation of the betweenness, the general algorithm will take \( O(mn^2) \) for \( m \) edges and \( n \) nodes, its cost for huge graphs is prohibitive.

Based on the breadth-first search algorithm, which finds the shortest paths from a source node \( s \) to all the others in time \( O(m) \), (Brandes, 2001) and (Newman et al.,
2004) have developed, independently, a method which takes \( O(mn) \) by finding the “leaf” nodes and then summing the weights of the edges from those nodes to a source node. Repeating the process for all possible nodes and summing the scores, gives the betweenness score for all the edges in the graph.

The fast unfolding algorithm, proposed by (Blondel et al., 2008), is an 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. 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.

After this first stage ends, a new graph is built whose nodes are the communities found in the first stage and the edges weight are the sum of the weights of the edges in the corresponding communities. Thus, the first stage is reapplied to this new graph and will iterate until no modularity improvement is possible. This algorithm is executed in linear time for sparse graphs ((Blondel et al., 2008)).

(Du et al., 2007) present an algorithm to detect communities in large–scale social networks. Their method is based on the enumeration of all the maximal cliques, i.e., a complete subgraph which is not contained in any other complete subgraph. After all the maximal cliques are enumerated, they generate kernels associated to those cliques and then, perform the community detection by assigning nodes to each kernel. After this, they try to optimize the modularity obtained by moving nodes accordingly. This algorithm has a complexity of \( O(\Delta \times M_C \times Tri^2) \), where \( \Delta \) is the maximal degree of the set of nodes, \( M_C \) is the size of the maximum clique, and \( Tri \) is the number of triangles present in the graph ((Du et al., 2007)).

Most of the classic algorithms will find disjunct partitions. However, several social networks from the real world may contain actors which belong to more than one community, i.e., overlapped communities. For example, (Pizzuti, 2009) presents a method for detecting overlapped communities. This method uses a genetic algorithm. The genetic algorithm’s fitness function is

\[
CS = \sum_{i \in S} score(S_i)
\]

where \( S \) is the set of groups, and \( score(S_i) \) is a quality measure which compares the edges within the group \( i \) versus the number of edges connecting nodes to the rest of the network.

(Lipczak et al., 2009) propose a genetic algorithm for detecting communities. In this case, the individuals are represented as a string of groups, which is a vector of size \( n \) containing the number of items in each of the \( n \) groups. During the selection and the
crossover operations, the genes are selected according to the potential improvement of the fitness function they may give. The fitness function, in fact, is composed of three measures, the normalized cut, proposed by (Shi et al., 2000), the modularity, proposed by (Newman et al., 2004), and the silhouette width, proposed by (Rousseeuw, 1987).

Other clustering methods, such as Markov Clustering, Iterative Conductance Cutting and geometric minimum spanning three, are discussed in (Brandes et al., 2008), and some methods for evaluating communities are presented by (Kwak et al., 2009) and by (Günter et al., 2003).

In general, to find communities all these methods only take into account the structural configuration of the graph: they do not include information associated to the nodes.

3. Defining the Point of View of Socio–Semantic Networks

Socio–semantic networks contain an important amount of information, coming not only from their topology, but from the different contexts in which such networks may be changed.

This information is associated to the actors and to the relationships in the network, and give more elements to analyze a network from different perspectives.

For example, for the actors form some network it is possible to use information related to the group they belong to, e.g., associations, enterprises, information about where they are located and academic backgrounds among others.

In the case of the relationships, it is possible to use information from the type of relation, e.g., familial, professional or friendship among others.

With this information it is possible to define two set of features, one for the actors set and one for the relationships set, which allow to make analysis of the network not from a structural perspective but from a semantic one.

It is also possible to use a subset of these features to analyze the network from different aspects, or points of view. This approach allows to extract knowledge by changing the point of interest to be exploited, i.e., by defining some type of multivariate filter from the feature sets.

The representation of the point of view and the association with the nodes will be explained in the next sections.

3.1. Some Notation

A social network can be represented as an undirected graph $G(V, E)$ where $V$ is the non-empty set of vertices, representing actors and $E$ is the set of edges representing the relationships among them.
Let $v_i$ and $v_j$ be two vertices from $V$ and let $e(x, y)$ be the edge defined by the vertices $x$ and $y$. Thus, if $e(v_i, v_j) \in E$ then $v_i$ and $v_j$ are neighbors.

Note that since $G$ is undirected, $e(x, y) \equiv e(y, x), \forall x, y \in V$. Note also that $e(x, x) \notin E, \forall x \in V$.

Given a graph $G$, a partition $C = \{C_1, C_2, \ldots, C_k\}$ is a partition of the set $V$ into $k$ non-empty disjoint subsets $C_i$.

Let $F_V$ be the set of features of the actors of the social network, which can be represented by a matrix of size $|V| \times |F_V|$. And let $F_E$ be the set of features associated to each edge, which can be represented by a matrix size $|E| \times |F_E|$.

Given a graph $G(V, E)$ and the features sets, a socio-semantic network $S$ can be defined as the tuple:

$$S = \langle G, F_V, F_E \rangle$$  \[1\]

With these definitions we will define the point of view.

### 3.2. Representation of Point of View

For the sake of this work, without lose of generality, we will define the point of view using the set of features of the nodes.

Given a semantic network $S = \langle G, F_V, F_E \rangle$, let $F^*_V \in \mathcal{P}(F_V) \setminus F_V$, where $\mathcal{P}(A)$ is the powerset of the set $A$, be a set of features to be used to define the point of view $PoV$.

Thus, for each vertex $v_i \in V$ there is assigned a binary vector $\xi_i$ of size $|F^*_V| = f$. If the vertex $i$ has the feature $p, 1 \leq p \leq f$ from $F^*_V$, then $\xi_{i,p} = 1$ or $0$, otherwise. Then, each binary vector $\xi$ can be defined as:

$$\xi_i = v_i \times F^*_V$$ \[2\]

where $v_i \in V$.

Then, a point of view is defined as the set of all instances derived from the set $F^*_V$. Using $2$:

$$PoV_{F^*_V} = \bigcup_{i=1}^{\left| V \right|} \xi_i$$ \[3\]
Figure 1 shows an example of how each node in the network is assigned to one instance $\xi$ of features. Note that different nodes could have the same instance $\xi$.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Feature 1</th>
<th>Feature 2</th>
<th>...</th>
<th>Feature</th>
<th>$}$ Instance $\xi_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Node 2</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Node n</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.** Example of the assignation of features to the network nodes.

It is also possible to define a point of view using information from edges. In this case, each edge contains information about the nodes composing it and its weight. Additionally, it may contain information about the type of the relation, e.g., friendship or familial.

Hence, instead of using the information from edges solely, it could be added to the point of view of nodes and thus obtain an enriched perspective of the semantic information.

Note that the generation of points of view from edges is not in the scope of this paper, however it is included in the project’s roadmap.

### 4. Using the Point of View to Influence the Clustering

The goal is to use both the structural information from the graph and the semantic information from the persons within the social network, including their relationships.

By using these information types, it is possible to guide the graph clustering process by adding information related to the similarity of the nodes in a real context. To do this, the community detection is divided into two phases. During the first one, the point of view is clustered using Kohonen maps ((Kohonen, 1997)) to obtain groups based on the similarity of the node features. Thus, the groups found in the first phase are used to change the weight of the edges in the graph, and then, in the second phase, a classic community detection algorithm is used. The diagram of the system is shown in the Figure 2.

The use of several phases for clustering has been used in different machine learning methods. By using preprocessing steps, the data used can be adjusted to be more suitable to certain algorithms. For example (González et al., 2006) train a SOM network to discriminate normal/abnormal samples from an artificial immune system and create the primary and secondary responses; (Deng et al., 2007) present a technique which first uses an hybrid clustering algorithm to create a neural fuzzy network, and then uses that network to predict diesel emissions.
4.1. Phase 1: Semantic Clustering

Given a point of view derived from a set $F^p_V$ defined as in Section 3, each node can be characterized by its vector of features, or an instance $\xi$ of the point of view.

It is possible to use those vectors as input training patterns in an unsupervised learning algorithm, such as Self-Organizing Maps (SOM) (Kohonen, 1997). This will group the nodes according to the similarities of their features, i.e., each instance $\xi$ is an input pattern for the training.

The SOM network $N$ has been implemented using a rectangular lattice of $l \times l$ neurons, where $l = |F^p_V|$ is the number of features in the point of view. The initial values of the weights of the SOM are randomly selected.

The weights of the neurons are adjusted using a learning rate $\eta$, which helps to avoid local optima. After each iteration this learning rate is reduced by a factor of $\epsilon$. The neighborhood is calculated using the size $\nu$ and the winner neuron $\omega$ as center.

The complexity of this algorithm is proportional to the number of features in the point of view, the number of instances and the size of the neural network. It can be expressed as:

$$T = O(f^3 \cdot n)$$ [4]

Where $n$ is the number of nodes of the graph and $f$ the number of features in the point of view.

The outcome of the algorithm is a partition $C_{SOM}$ (recall partition definition from Section 3.1) of the nodes assigned to the neurons.
4.2. **Phase 2: Structural Clustering and Community Detection**

Once the semantic partition $C_{SOM}$ has been found it is possible to begin the second phase of the proposed method.

During this phase we use a classic graph clustering algorithm to find communities, specifically, the fast unfolding algorithm, proposed by (Blondel et al., 2008) and presented in Section 2. This algorithm uses the modularity $Q$, presented by (Newman et al., 2004) as quality measure.

Before the execution of the fast unfolding algorithm, we include the information from the phase 1. This is performed by changing the weights of the edges according to the partition $C_{SOM}$.

Thus, for each pair of neighbor vertices $v_i, v_j, \forall i \neq j \in V$, the weight of the edge $e(v_i, v_j)$ is changed according with the Euclidean distance of the PoV instances corresponding to each node by:

$$w_{ij} = 1 + \alpha (1 - d(N_{ij})) \delta_{ij}$$  \[5\]

where $\alpha \geq 1$ is a constant value, $d(N_{ij})$ is the distance of the neurons $i$ and $j$, and $\delta_{ij} = 1$ if $v_i$ and $v_j$ belong to the same partition in $C_{SOM}$, $\delta_{ij} = 0$ otherwise.

The algorithm 1 shows how the results from semantic clustering are merged into the graph $G$.

**Algorithm 1** Weight assignation algorithm

```plaintext
for each $e \in E$ do
    $i \leftarrow 0$
    $found = false$
    do
        if src($e$) $\in c_i \land$ target($e$) $\in c_i$ then
            adjustWeight($e$) (Eq. 5)
            $found = true$
        end if
        $i \leftarrow i + 1$
    while not $found \land i < |C_{SOM}|$
end for
```

After the weights are changed according to Equation 5, a partition $C_{SOM-FU}$ is found using the Fast Unfolding algorithm.

This new partition $C_{SOM-FU}$ contains the final set of communities, which has both, the semantic information and the structural information.
Since our approach adds a preprocessing layer in order to find a semantic partition \( C_{SOM} \), the complexity is given in function of the complexity of the semantic clustering plus the complexity of the fast unfolding algorithm.

The complexity of the semantic clustering is given by Equation 4 and the complexity of the fast unfolding algorithm has been reported to be linear in the number of nodes for sparse adjacency matrices ((Blondel et al., 2008)), i.e., matrices having more zeroes than ones.

Thus, since the social networks are in general sparse, (Boukli, 2006), the overall complexity of our method for a defined \( F_{\Psi}^{*} \) is \( O(|F_{\Psi}^{*}|^3|V|) \).

5. Preliminary Experiments

Some preliminary experiments were developed using three graphs and points of view randomly generated. The table 1 shows the configuration of the generated graphs. The density measure is calculated using:

\[
\delta = \frac{2 \times |E|}{|V| \times (|V| - 1)}
\]

where \(|E|\) is the number of edges and \(|V|\) is the number of nodes. Hence the maximum number of edges in a graph of \( n = |V| \) nodes is \( \frac{n \times (n-1)}{2} \), the maximum density \( \delta \) is 1.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Nodes</th>
<th>Edges</th>
<th>Density</th>
<th>Initial Modularity</th>
<th>PoV</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>3892</td>
<td>0.2</td>
<td>-5.1216 \times 10^{-9}</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>5389</td>
<td>27347</td>
<td>1.8836 \times 10^{-9}</td>
<td>-2.5192 \times 10^{-3}</td>
<td>2</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 1.** Description of the graphs and points of view used during the experimentation. The graph 1 was artificially generated. The graph 2 is an extract from Twitter. The points of view 1 was created randomly, and the points of view 2 and 3 were generated extracting data from the Twitter data set.

The graph 1 was artificially made to test the method in a general way. The graph 2 is an extract from Twitter, which can be seen as a real network. Note that the density of the artificial graph is higher than the density of the real ones.

For the experimentation we compare the clustering results of two classic clustering algorithms and our proposed method. The first classic algorithm is the SOM, it finds partitions based only on the semantic information, denoted by \( C_{SOM} \), the second classic algorithm, fast unfolding, it finds communities using only the structure of the graph, denoted by \( C_{FU} \), and our proposed method denoted by \( C_{SOM \rightarrow FU} \).
To measure the result of the experiments, we use the average Euclidean distance within the groups obtained calculating for each pair of nodes from a group, the distance between the instance of the point of view assigned to each one and the modularity $Q$ to evaluate, from a structural perspective, the obtained partition.

The points of view used in the experiments are:

1) **Random point of view**: a point of view with five features randomly generated. There are $2^5$ possible instances with the same assignment probability. Note that an instance could be assigned to more than one node.

2) **Time zone division**: it is composed of 33 features representing the different time zones registered in the data set. The time zones are the difference in seconds with the Prime Meridian. Thus, each instance is described by the presence of friends in each time zone.

3) **User profile**: The first feature indicates if the user follows more people, or in the twitter environment, has more friends, than followers. Users who have more followers than friends are usually people, or organizations, which have a lot of people interested in their updates and messages. This is the case of politicians and public figures. The next three features indicate the user behavior according to the number of messages sent. Thus, the features are: below the mean, between the mean plus three standard deviations and, over mean plus three standard deviations. In this data set nearly 82% of users are below the mean of the messages sent.

5.1. **Experiment 1 – Graph 1 with PoV 1**

The first experiment was executed using a graph of 200 nodes and 3892 edges. In this experiment, the groups found by the fast unfolding algorithm have a higher average intracluster distance. Note that the greater the distance between two nodes, the greater dissimilarity is regarding the features of the point of view.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Final $Q$</th>
<th>Average Intracluster Distance</th>
<th>Standard dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{SOM}$</td>
<td>-0.0056</td>
<td>0.2793</td>
<td>0.2370</td>
</tr>
<tr>
<td>$C_{FU}$</td>
<td>0.1885</td>
<td>1.5375</td>
<td>0.4117</td>
</tr>
<tr>
<td>$C_{SOM\rightarrow FU}$</td>
<td>0.5389</td>
<td>1.0833</td>
<td>0.5692</td>
</tr>
</tbody>
</table>

Tableau 2. Results of the experiments performed with the graph 1, comparing the result of the classic algorithms versus the proposed method.

The preliminary results in Table 2 show that the average distance of the nodes inside the groups found by our method is less than average distance for the groups found by the graph based algorithm.

This means that from the semantic perspective, the groups found by the proposed method are more similar. Additionally, the modularity of the partition is higher because of the change of the weights in accordance with the semantic groups.
Additionally in the first row of the table the results of a graph created directly from the SOM execution are presented. In this case, the modularity is even worst than the original graph but, the average distance, as we expected, is small.

This experiment shows how the groups found from the point of view can influence the results of the classic graph clustering algorithm. In this case the modularity has been improved in an important way, though the intracluster similarity tends to be preserved.

5.2. Experiment 2 – Graph 2 with PoV 2

This experiment was executed using a graph of 5389 nodes and 27347 edges extracted from a Twitter data set, composed of ∼ 204000 nodes and ∼ 326000 edges.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Final $Q$</th>
<th>Average Intracluster Distance</th>
<th>Standard dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{SOM}$</td>
<td>-0.0075</td>
<td>0.3697</td>
<td>0.1059</td>
</tr>
<tr>
<td>$C_{FU}$</td>
<td>0.5728</td>
<td>1.8091</td>
<td>1.3584</td>
</tr>
<tr>
<td>$C_{SOM \rightarrow FU}$</td>
<td>0.5747</td>
<td>1.1947</td>
<td>0.8489</td>
</tr>
</tbody>
</table>

Tableau 3. Results of the experiment performed with the graph 2, comparing the result of the classic algorithms versus the proposed method using the point of view 2.

In this experiment, reported in Table 3, the average intracluster distance found by our proposed method is less than the average intracluster distance found by the graph based algorithm.

In both executions the modularity obtained is very similar. This is due to the structure of the point of view, which uses information associated with the localization of people’s friends. We may think here that friendship tends to be similar when considering friends.

In the next experiment we use a different point of view which is not related to the way people create relationships in the network. We characterize them by their interaction profile.

5.3. Experiment 3 – Graph 2 with PoV 3

This experiment was executed using the same graph of the experiment 2, but with the point of view 3. Results are shown in the Table 4.

The point of view used can be expressed according with the first feature, which may take two states, present or not and the other three are disjoint, so if one of them is set to one, the other two are zero. Thus, there are $2 \times 3 = 6$ different possible instances.
Tableau 4. Results of the experiment performed with the graph 2, comparing the result of creation of community graph from the SOM, the classic clustering algorithm and the proposed method using the point of view 3.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Final $Q$</th>
<th>Average Intracluster Distance</th>
<th>Standard dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{SOM}$</td>
<td>-0.2991</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{FU}$</td>
<td>0.5728</td>
<td>0.7100</td>
<td>0.6565</td>
</tr>
<tr>
<td>$C_{SOM\rightarrow FU}$</td>
<td>0.6351</td>
<td>0.5507</td>
<td>0.5577</td>
</tr>
</tbody>
</table>

The SOM clustered the nodes into six groups, each one expressing one of the possible instances described above. This explains the average distance found.

Creating a graph from the SOM clustering will produce better semantic clusters, however, the modularity is worst than the one from the original graph. This shows that the SOM groups are totally unrelated with the structure of the graph.

In the case of the graph based clustering and the PoV based clustering the results are different. The performance of the PoV based algorithm was better according to the modularity and the average intracluster distance.

6. Conclusion and Future Work

The information contained into a socio-semantic network is tied to certain features of the actors and the edges. Such information allows to perform analyses over the network from different perspectives, i.e., from different points of view.

The classic community detection algorithms use information only from the network structure and do not take into account the semantic information, which could be used to influence the clustering process.

Assigning the weights derived from the results of the semantic clustering to the edges, the semantic information is included into the community detection process and the two types of informations are merged to find and visualize a social network from a selected point of view.

Regarding the execution time of our method, the complexity is higher than the complexity for the graph based. Today, this imposes some restrictions in the number of features. The sensibility of the execution time to the number of features is high because of the SOM training.

The high number of dimensions may mislead the SOM training because of the Hughes effect ((Hughes, 1968)), also known as the curse of dimensionality, and how the semantic distance is measured. Hence, we will study the statistical properties of the points of view to try to reduce this effect.
For future work we will also continue the study of the influence of the point of view in the community detection process including the definition of points of view from the graph’s edges. Additionally, we plan to work on the development of a visualization algorithm for hierarchical social networks.

7. Bibliographie


Bouklit M., Autour du graphe du Web : Modélisations Probabilistes de l’Internaute et Détecti


