Information Flow using Edge Stress Factor

Franco Salvetti
University of Colorado at Boulder
430 UCB
Boulder, CO 80309-0430 USA
franco.salvetti@colorado.edu

Savitha Srinivasan
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120 USA
savitha@almaden.ibm.com

ABSTRACT
This paper shows how a corpus of instant messages can be employed to detect de facto communities of practice automatically. A novel algorithm based on the concept of Edge Stress Factor is proposed and validated. Results show that this approach is fast and effective in studying collaborative behavior.

Categories and Subject Descriptors
G.2.2 [Discrete Mathematics]: Graph Theory – graph algorithms.

General Terms: Algorithms, Experimentation.

Keywords: Social Network Analysis, Graph Clustering.

1. INTRODUCTION
Community detection, as part of the general problem of expertise location, has become an important aspect of knowledge management systems. In modern organizations expertise sometimes is spread over a group of people, and therefore knowing which group is doing what is crucial for finding someone to help with a particular task. People frequently collaborate with others who are not part of their "official" team, and organize in spontaneous communities. Consequently, community detection solutions based on "official" information are in general incomplete and not cost effective. Corporate users have discovered that instant messaging helps them exchange small but often critical details; hence, a corpus of instant message logs (MLog) contains enough information to become a useful asset that can be exploited for community detection. The graph implicitly defined by an MLog, where a node is a person and an edge represents an instant message exchange, is expected to have community structure [1] as in Figure 1, which shows sets of nodes densely connected internally, but with lower density of external links. This paper presents and validates an efficient iterative algorithm capable of finding such highly interconnected subsets of people by leveraging local information flow to define an *Edge Stress Factor* (ESF).

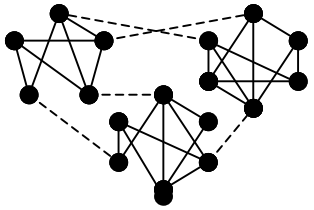

Figure 1. Graph with community structure.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

2. RELATED WORK
In Social Network Analysis (SNA) nodes in a graph are actors, while edges represent relationships [2]. SNA has been used to identify structures in social systems (e.g. terrorist networks) based solely on the relations among actors. Some methods used to analyze networks are based on the centrality (i.e. influence, or prominence) of nodes and edges in a network. In [2] an algorithm is proposed which involves iterative removal of edges with high centrality from the network to split it into communities. Other relevant research can be found in the areas of hyperlink network analysis, data mining and collaborative filtering.

3. INFORMATION FLOW
The focus is on defining the information flow measure that best captures the communities implicit in an MLog. Different measures of centrality of a node or edge, such as betweenness, closeness, degree, and information, capture aspects of its role in a network. Among them the betweenness centrality (BC) captures the degree of influence a node has over the flow of information in a network.

3.1 Betweenness Centrality
Roughly speaking the BC of a node is the number of shortest paths between pairs of nodes that pass through it [2]. Likewise, the BC of an edge can be computed as well [1]. A high BC suggests that an edge is likely to bridge two communities.

3.2 BC for Community Detection
A baseline algorithm [1] that uses the BC to detect communities from a graph G of an MLog can be stated as follows:

1. Compute the BC for all edges in G.
2. Remove the edge with the highest BC.
3. Recalculate BC for all edges involved in the removal.
4. Repeat from step 2 until no edges remain.

Given a node x, the addition of a new community "far" from x in G could drastically modify the BC of the edges "around" x. This appears counterintuitive because it can lead to "separation" of nodes which should belong in the same community. Moreover in this framework it is not possible to use weights (e.g. number of conversations), and step 3 potentially recalculates the BC of all edges. For these reasons the flow measure ESF is introduced.

3.3 Edge Stress Factor in Information Flow
The goal is to remove edges iteratively in G and to detect communities arising from this process. Consider the edge (a, b) in G and all the edges (k_i, a) and (h_j, b) such that k_i is connected with a but not with b, and h_j is connected with b but not with a as in Figure 2.

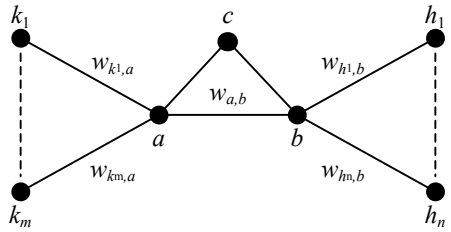

Figure 2. The ESF of (*a*, *b*) is computed as in (1).

Metaphorically a source nodes h_j is trying to communicate a certain amount of information $w_{hj,b}$ to the destination a, and to do so it "stresses" the edge (b, a) with "capacity" $w_{a,b}$, by passing through the mediator b. At this stage we are not considering the presence of multiple mediators. The ESF is computed as in (1) and provides a way to measure local information flow.

$$ESF_{a,b} = \left(\sum_{i=1}^{m} w_{k_i,a} + \sum_{j=1}^{n} w_{h_j,b} \right) - w_{a,b} \qquad (1)$$

It is intuitive that the ESF will be higher on inter community than intra community edges, because the number of "missing" edges is lower within a community than between two communities.

3.4 Local Information Flow

Recalculating the BC after the removal of an edge is computationally expensive, but if an edge which is between communities "globally" is also between them "locally", it is possible to use the ESF instead of the BC in the previous algorithm. When an edge is removed, to recalculate the ESFs it is possible to add the weight of such an edge to all the edges previously "stressed" by it. Therefore, the notion of local information flow dramatically reduces the computational cost.

3.5 Algorithm to Detect Communities

The MLog becomes an undirected graph, where each node represents a person, and each edge a conversation. Each edge carries a weight corresponding to the number of messages exchanged between the two persons. In order to define a criterion for accepting as a community a connected component generated through iterative edge elimination, the following are employed:

***k*-core**: a subgraph Q with n vertices is a k-core if any internal node is adjacent to at least k nodes in Q.

***k*-core-factor**: the k-core-factor of a subgraph Q is the maximum k for which Q is a k-core.

community: A subgraph Q with n vertices and a certain k-core-factor is accepted as a community if k is greater or equal to $\alpha \cdot (n\text{-}1)$ with α in [0,1].

The final version of the algorithm after each edge removal verifies whether a new connected component is present, and whether such a component matches the previous definition of "community" for a given value of α. In such a case it returns the component as a community and eliminates it from the original graph.

4. COMMUNITY DETECTION

Two datasets were available: the first containing 350 corporate instant messages contributed by 100 users, the second ~220,000 of IBM's global intranet instant messages produced in about two hours by ~100,000 employees.

4.1 Validation Methodology

Subject matter experts organized the users in the first dataset into 5 distinct communities, then compared them with the algorithm's results. For the second dataset after the aggregation of the data by country, division and work location, the experts judged the appropriateness of the extracted communities.

4.2 Experimental Results

Among the results, Figure 3 shows the users' connectivity graph aggregated at the division level. The algorithm with α=0.7 identifies the four communities (A, B, C, D) which correctly represent Sales, Integrated Supply Chain, Server Brand Management, and Storage Technology.

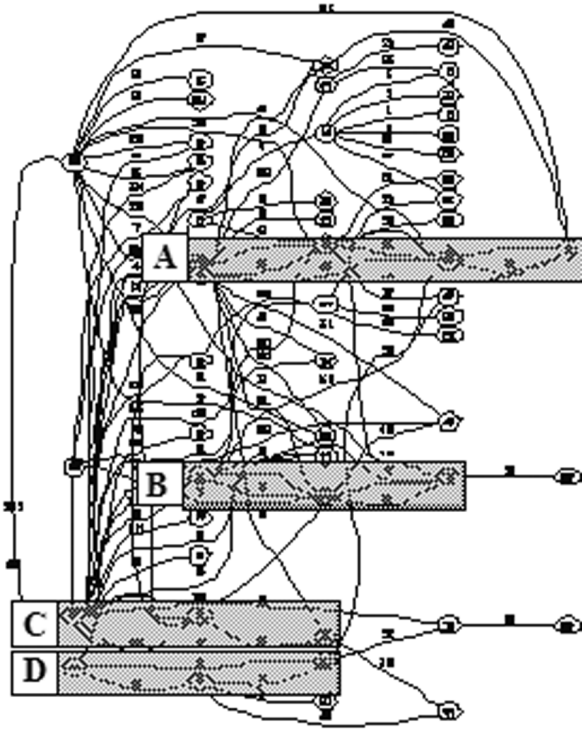

Figure 3. Four communities in the division level graph.

The communities detected in the first dataset and in the different aggregations of the second were meaningful, showing the ability of the algorithm to extract communities from social networks.

5. CONCLUSION

A novel algorithm based on computing an Edge Stress Factor has been proposed to extract communities from graphs implied by an instant messages corpus. The algorithm leverages an idea of local information flow, which is formally modeled as the Edge Stress Factor. The locality of the algorithm allows the use of weights in the graph and reduces computational cost. Results have been successfully validated against real data.

6. REFERENCES

[1] Newman, M. E. J. and Girvan, M., Finding and evaluating community structure in networks, Phys. Rev. E, in press, 2003.

[2] Wasserman, S. and Faust, K., Social network analysis: Methods and applications. Cambridge, NY: Cambridge University Press, 1994.