SIGHTS: A Software System for Finding Coalitions and Leaders in a Social Network

J. Baumes*, M. Goldberg†, M. Hayvanovych†, S. Kelley†, M. Magdon-Ismail†, K. Mertsalov† and W. Wallace‡
*Kitware, Inc., Email: jeffbaumes@gmail.com
†CS Department, RPI, Rm 207 Lally, 110 8th Street, Troy, NY 12180, USA.
Email: {goldberg,hayvam,kelles,magdon,mertsk2}@cs.rpi.edu
‡DSES Department, RPI, 110 8th Street, Troy, NY 12180, USA.
Email: wallaw@rpi.edu

Abstract—We present an extended version of a software system SIGHTS[1] (Statistical Identification of Groups Hidden in Time and Space), which can be used for the discovery, analysis, and knowledge visualization of social coalitions in communication networks such as Blog-networks. The evolution of social groups reflects information flow and social dynamics in social networks. Our system discovers such groups by analyzing communication patterns. The goal of SIGHTS is to be an assistant to an analyst in identifying relevant information. The functionality of SIGHTS includes:

- Discovery of coalitions (clusters) and their leaders;
- Finding hidden groups using communication persistence techniques;
- Discovering hidden groups in communication streams;
- Matching topics of the blogs and detecting sentiments;
- Tracking the evolution of clusters;
- Visualization of collections of individual clusters.

Keywords: Data Collection; Learning; Knowledge Visualization.

1 Introduction and Motivation

Modern means of communication, such as e-mail, web-logs, and chatrooms, allow individuals to communicate in a number of new ways, and new forms of communication are continually appearing. This vast and ever growing digital communication data must be processed in order to extract information relevant to an analyst. Information crucial for analysis concerns social coalitions and their evolution. While some coalitions publicize their presence in the network, the intention of others is to hide their communication, and their existence within the large body of all communication in the network. Yet for some groups, their members are not even aware of the existence of those groups to which they belong.

Social network analysis (see [2], [3], [4], [5], [6], [7]) can be used to understand the social dynamics caused by major events such as unrest in Muslim communities around the world created by the publication of cartoons in a Dutch newspaper in September of 2005. Information about this publication spread through social networks reaching countries that are geographically distant from the origin of publication.

The main objective of SIGHTS is to provide an umbrella for various algorithms for use in analyzing communication data with the goal of detecting social coalitions, primarily, "hidden" groups. The graphical user interface of SIGHTS contains three facilities to help the analyst examining and manipulating the results of the algorithms.

The advantages of this approach include the following.

1) The algorithms can be used by several users.
2) Increased use provides increased feedback for future improvements.
3) Placing all strategies in a single place may uncover more effective combined strategies or a synergy between strategies.

The three main modules of SIGHTS (see Figures 1 and 2) are: Data Collection/Storage, Data Learning and Analysis; and Knowledge Extraction/Visualization. The Data Collection/Storage

1. The first version of SIGHTS was developed by J. Baumes, M. Goldberg, M. Magdon-Ismail, and W. Wallace, based on the results presented in [1].

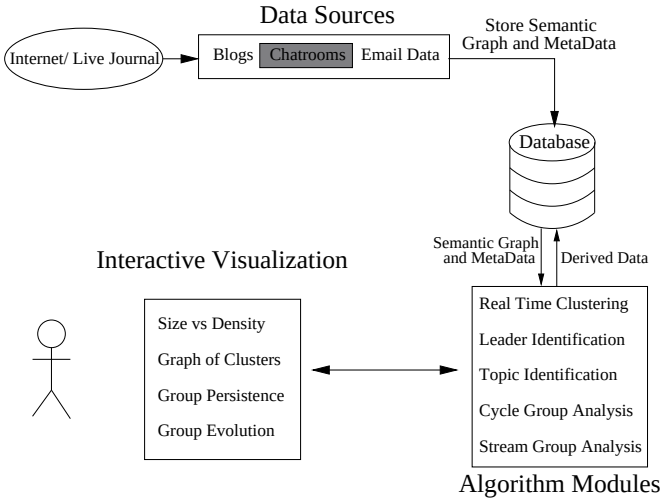

Fig. 1. SIGHTS System Architecture(*currently the link from Chatrooms is not functional*)

module contains multiple data sources, each responsible for gathering data and representing it as a semantic graph for further processing. Currently we have developed modules that collect data from LiveJournal.com blogs, Enron email corpus, as well as random communication data generators. The resulting semantic graph is stored in the database where it can be found by the data processing facilities that help extract information such as identifying overlapping clusters representing social groups; finding group leaders; finding hidden groups using cycle and stream models; and tracking the group evolution. Data collection and data processing can be run in parallel and the collected data analyzed in real time. At any point the analyst may look at the data using the interactive visualization module. It presents data through multiple visualization facilities, including the *size vs. density*-plot for group analysis; *clusters* in the graph, and *groups in time and group evolution*-plots. Since the visualizations are interactive, they allow the analyst to zoom into a certain part of the visualization and select groups that are of particular interest; the system then provides additional information regarding the group in question.

Some of the algorithms used in SIGHTS are described in our early publications, while others are in preparation. The latter include the algorithms for topic identification, leader discovery, and tracking group evolution.

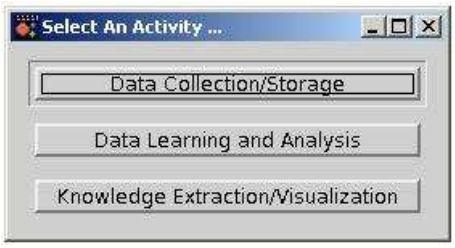

Fig. 2. SIGHTS Startup Window.

ACKNOWLGEMENT

This material is based upon work partially supported by the National Science Foundation under Grant No. 0324947 and by the ONR Grant No. N00014-06-1-0466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or U.S. Government.

2 DATA COLLECTION MODULES

The system operates on semantic graphs. The graphs are constructed by adding a node for each social network actor and a directed edge from sender's node to a recipient's node. The edges are marked with the time of the communication and, possible, other labels. Some edge labels are only appropriate for specified types of graphs.

2.1 Loading a Stand-alone Dataset

The user may have communication data existing in a variety of formats. SIGHTS handles the stand-alone input of a reasonable range of these formats in order to facilitate the introduction of new data into the program. Among these is a plain-text XML format which is well-documented. SIGHTS is also able to read from a database that is constructed according to specified guidelines.

2.2 Blogs data collector

Blogs data is collected from LiveJournal.com blogs service provider. LiveJournal.com provides the fertile ground for social interaction and creation, and evolution of communities thought unlimited depth comment threads, friendship links declaration and explicit community declaration. The semantic graph is constructed by creating a node for each blogger and the edge between any pair of bloggers who participated in the discussion in the comments of a post.

Blogs collector monitors LiveJournal.com update feed and records the permanent address of the post. Two weeks after the date of the initial post, the blogs collector visits the page of the post and collects the thread of comments using the screen-scraping techniques.

Blogs collector allows the establishment of "interest filters" that can narrow the data collection to posts on a certain topic. This allows the analyst to focus on a particular portion of the communication network that occurs in Blogs and only pertains to a certain topic. The filters are set up using the supervised machine learning technique based on the training provided by the analyst. The training can be done via (a) a manual review of selected posts; or (b) the articles that are posted on the external resource and are accessible through an RSS-feed. Blogs collector provides the interface for the analyst to tag posts as interesting and not interesting that will create the training set for the interest learning program. Alternatively, analyst can identify the RSS-feeds that provide the articles of the topic interesting to the blogger. The articles posted on these feeds will be used by the filter to learn the topic interesting to a user. The simplest way to find such feed is to search on Google. For example, searching for term "RSS business news" results in hundreds of RSS-feed that serve business related data. Any or all of these feeds can be used to train the system to tag "business" posts. The system also needs examples of articles that are not interesting to a user. Similarly, the user can find the RSS-feeds that post the data of no interest. From these articles the filter will learn what user is not interested in. The feeds can be found just the same way on Google. Simple search for "rss news" on goggle resulted in a link to Yahoo RSS news page (http://news.yahoo.com/rss) that lists number of feeds in various categories. The categories that are not interesting to a user will make up the negative set. With enough training data the filer tags semantic graph edges as interesting and not interesting. This information is also stored in the database and is accessible to other modules of the application.

2.3 Generating a Simulated Dataset

The features of the tool and the correctness of the algorithms can be assessed by generating, within SIGHTS, a synthetic data set. For the background communications, the user is able to select random graph $G(n,p)$, where p is the probability of a message per unit time. Another option is random $G(n,p)$ with static casual groups with sizes selected randomly from a specific size distribution. The activity of these casual groups is simulated by overlaying $G(n_i,p')$ on top of the uniform background, where $p' \gg p$. These casual groups communicate much more frequently than a random set of actors, but the connectivity of these actors is not enforced. The user may also simulate a preferential attachment model for background communication.

In addition to the background, the user is able to embed hidden groups. The structure of the hidden group may be random, independent connectivity over fixed or variable cycle lengths (i.e. random spanning trees for each cycle). The hidden group may also have a fixed structure such as a tree, with a specified number of levels, and propagation delays which may be constant or chosen from a random distribution.

2.4 Enron Corpus

The Enron email corpus consists of e-mails released by the U.S. Department of Justice during the investigation of Enron. This data includes about 3.5 million e-mails sent from and to Enron employees between 1998 and 2002. The corpus contains detailed information about each email, including sender, recipient(s) (including To, CC, and BCC fields), time, subject, and message body. We converted it into the semantic graph that can be analyzed using data processing and visualization tools provided by SIGHTS.

3 Data Processing Modules

The semantic graph and meta data obtained by the Data Collection module can be processed with data processing modules to retrieve additional information. Processing modules will run in parallel with data collection and the identified groups and other data will be stored in the database. Using visualization module user may view the data.

3.1 Temporal Group Algorithms

The following algorithms identify hidden groups in the stream of communications. Although we do not provide the detailed analysis and explanation of the Cycle and Stream algorithms, however all the details and previous work can be found in [8], [1], [9], [10].

The temporal hidden group algorithms (cycle and stream models) require that the user specify

the time scale for analyzing the data. The user is able to split the time line into cycles with specified constant duration or constant number of messages in each cycle. The user may also select the "phase" of the cycles, i.e. the time at which the first cycle begins.

3.2 Cycle Model

The user may run a cycle model algorithm to generate a database of all possible internally persistent groups in all ranges of cycles. See an example of an internally persistent group in Figure 3. User may configure start and end points of the analysis and the cycle duration length. As groups are found they are stored in the database. Using the visualizations user may extract groups that are of interest.

3.3 Stream Model

The user is also able to run the stream algorithm from SIGHTS. This algorithm finds groups based on frequent triples and siblings and by clustering them into larger groups allows for tracking the evolution. See an example of the evolution of a group found in the ENRON email data set in the Figure 4.

SIGHTS allows a user to configure the various parameters used for processing under the stream model. This includes start and end points of the analysis, cycle duration as well as time interval bounds, thresholds for minimum number of siblings and the threshold used during the clustering.

3.4 Overlapping Clustering Module

The user may cluster a specific set of actors, using communications collected over a specific interval of time. The details of this algorithm are presented in [11], [12]. SIGHTS then provides the user with an intuitive way to interact with the results. The groups may be plotted on a size vs. density plot, where the user may click on a group to view its members. Specifically, the visual result must highlight the overlaps among clusters.

In addition, SIGHTS contains an algorithm for discovering overlapping clusters that evolve over time. The algorithm first independently clusters each communication cycle for a specific set of cycle boundaries. The program then heuristically matches clusters in order to produce the evolution of clusters over time.

Another approach would be to construct one large graph which is the union of graphs

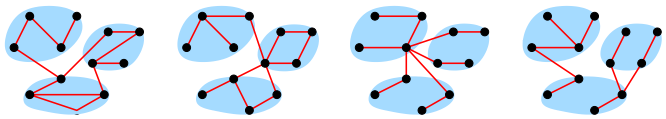

Fig. 3. Internally Persistent Group Example

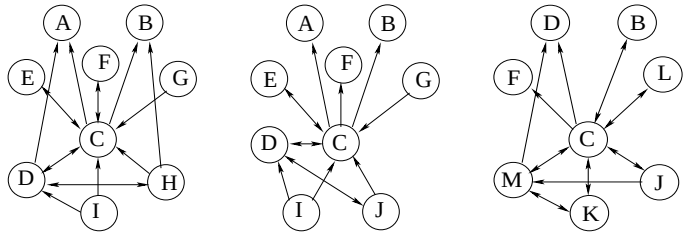

Fig. 4. Evolution of a part of the Enron organizational structure from 2000 - 2002. All three graphs represents frequent structures; the labels indicate actors of the Enron community. Note: that B, C, D, F are present in all three time-intervals.

$G_1, G_2, \ldots, G_n$ with nodes repeated for each interval. One would then add edges linking nodes corresponding to the same actor. Clustering this large graph could yield groups which are allowed to change over time. Experimenting with this second approach, two drawbacks were found. The first is that the entire dataset must be loaded at one time, which is an impractical memory requirement for large datasets. Second, the evolutions discovered by this method tended to have "spurs" consisting of single messages radiating from the group which were difficult to interpret.

3.5 Opposition Identification Module

This facility identifies the positive and negative sentiments between pairs of bloggers based on the length and average size of the messages in the conversations that took place among the bloggers.

The threads of comments that appear on LiveJournal.com are split into conversation between pairs of bloggers. Conversation is a continuous mutual exchange of messages between two bloggers that appear in the same thread. Typically the bloggers who argue engage into conversations that contain many messages and the average message length is longer. The module employs the Support Vector Machine model that was trained using a data set that was manually created to determine the oppositions between bloggers using the length of the conversation and the average length of the message

in the conversation to determine whether bloggers opposed each other in a given conversation. The information about the conversation sentiment is stored in the database and is used by the clustering module for more precise results.

4 Interactive Visualizations

System SIGHTS provides a GUI for interactive visualization of the results obtained from data collection and data processing modules. Currently system supports four different visualizations for the analyst: Size vs. Density plot, graph of Clusters plot, Group Persistence view and Group evolution view. There are four main parts to each visualization: the interactive plot, time range selector, the selected group detail window and the list of analysis available on a current graph. As shown in Figure 5, the list of available analyses can be found in the upper left corner of the main window. The `''Selected Group''` display is located right below it and the time range selector is located in the lower left side of the main window.

4.1 Size vs. Density Plot

The interactive plot in Figure 5 displays the groups (each group is represented as a dot) discovered in a cycle within the dates given by the range selector bars. Dots are placed on the screen according to their size (given by the label on the top of the plot) and their density (given by the label on the left of the plot). Density is simply a measure of the communications among individuals in the group against communications to individuals outside of the group.

The shading of the squares indicates their interest level. Bold dots correspond to groups with a high level of communications which have been marked as interesting based on the filters set up during the configuration of the data collection phase.

4.2 Graph of Overlapping Clusters

This interactive plot as shown in Figure 6 displays the groups discovered in the time range determined by the range selector bars. In this view each grey dot represents an actor and grey links represent the background communications between actors if such exist. Every group is denoted as a green square, and the links from the green square to grey dots show which actors are members of this group. Clicking on a group will make it selected, and the

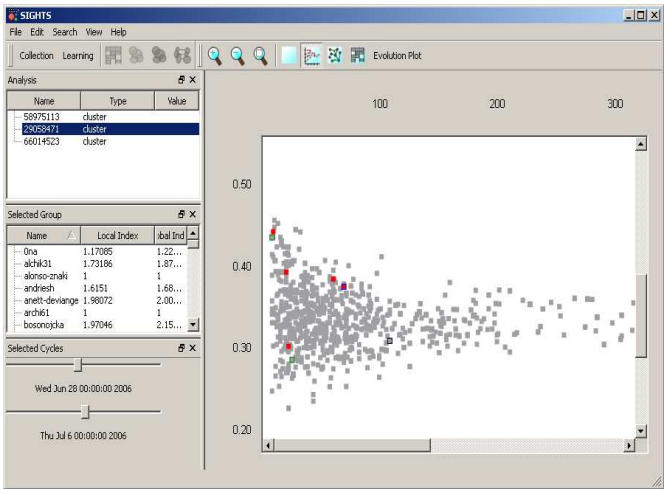

Fig. 5. Size vs. Density Plot of SIGHTS Sample Analysis (*Each dot represents a group, bold dots are groups of high interest level with high amount of communication*)

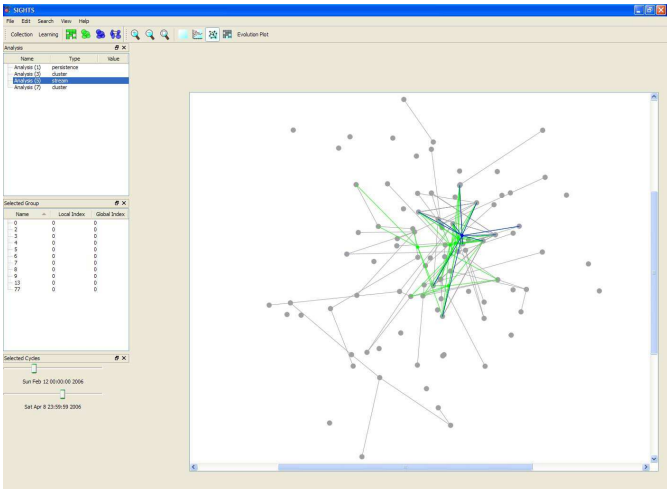

Fig. 6. Graph Clusters Plot of SIGHTS Sample Analysis (*Grey dots are actors, each green dot with links to actors represents a group and grey links correspond to background communications*)

`''Selected Group''` display on the left of the interactive plot will show the information about the members of this group.

4.3 Visualization of Group Persistence

This visualization gives the analyst an opportunity to view all of the groups over all of the time cycles, which are represented by vertical lines. Each time cycle has a number of rectangles which belong to this cycle and represent groups. Once again, when a group is selected, its members will be shown in the `''Selected Group''` display. While the

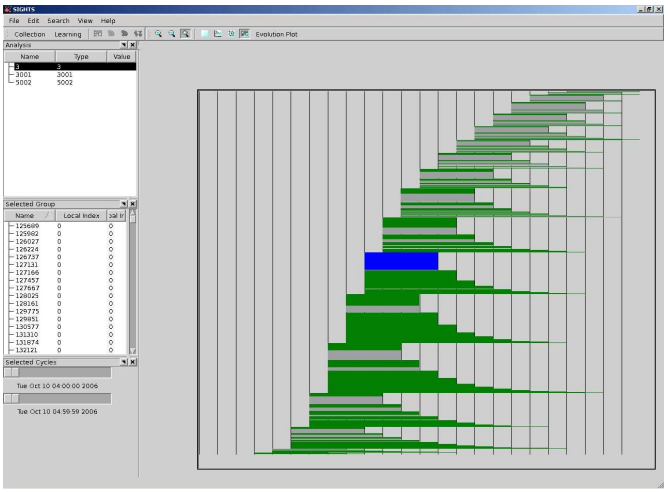

Fig. 7. Group Persistence Plot of SIGHTS Sample Analysis (*Vertical lines define the time cycles, each rectangle is a group. A selected rectangle is entirely blue, other rectangles are grey if they have no members in common with selected one, partially/entirely green if they have some members in common or partially/entirely blue if a group contains all of the members of the selected rectangle. If the rest of the rectangle is grey it indicates that it has some additional members*)

selected group is entirely colored blue, other groups can be either colored grey, which means they do not have any members in common with selected group, or a group can be partially/entirely colored green or blue, which respectively means that it contains some but not all of the members from the selected group, or has all of the members of the selected group as well as in both cases (green or blue) groups can possibly have some other actors in them. Those actors will correspond to the remaining portion of the rectangle and will be colored grey. An example of the group persistence visualization can be found in the Figure 7.

4.4 Leaders and Groups Evolution

When a square is clicked on in the interactive plot, the analysis will be updated with two important pieces of information. On the left of the interactive plot is a display labeled "`Selected Group`" which upon clicking will be updated with all of the members of the group. This display will present three portable columns showing the blogger's name, global leadership index, and local leadership index. Double clicking on any of the entries of this display will open a web browser and navigate to the selected user's blog.

Leaders and Group evolution view displays leadership information and group members. Clicking on a coalition will find it's evolution. The currently selected group will be outlined in blue. Any coalitions that act as ancestors to the selected group will be outlined in green if they were discovered in a cycle that is currently displayed in the plot. Descendants of a selected coalition will be outlined in black if they were discovered in cycles currently displayed in the plot. Clicking on a descendant cluster will update the plot and allow the analyst this way to track the evolution.

5 System Implementation

System SIGHTS uses Postgres database to store all collected data. Data processing modules access the database for semantic graphs and use the database to store the result of data analyses such as overlapping clusters, group leader and group evolution information.

Data collection and processing modules run in parallel. They are managed by simple scheduling system that keeps track of the state of processes and activates new processes. System SIGHTS includes the visual web-based interface that allows administrator to manage and configure the processes that are currently running.

All modules except the visualizations can be written in any language that can connect to Postgres database. The modules of the current implementation are written in C++ and PHP5.

References

[1] J. Baumes, M. Goldberg, M. Magdon-Ismail, and W. Wallace, "Identifying hidden groups in communication networks," *to appear in Handbooks in Information Systems – National Security*, 2005, invited book chapter.

[2] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[3] T. Berger-Wolf and J. Saia, "A framework for analysis of dynamic social networks," *DIMACS Technical Report*, vol. 28, 2005.

[4] J. Sinai, "Combating terrorism insurgency resolution software," *IEE International Conference on Intelligence and Security Informatics (ISI-2006)*, pp. 401–406, 2006.

[5] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2006, pp. 44–54.

[6] D. Kempe, J. M. Kleinberg, and E. Tardos, "Influential nodes in a diffusion model for social networks." in *ICALP*, 2005, pp. 1127–1138.

[7] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2003, pp. 137–146.

[8] J. Baumes, M. Goldberg, M. Hayvanovych, M. Magdon-Ismail, and W. A. Wallace, "Finding hidden groups in a stream of communications," *Proc. IEEE International Conference on Intelligence and Security Informatics (ISI-2006)*, vol. 3975, pp. 201–212, 2006.

[9] J. Baumes, M. K. Goldberg, M. Magdon-Ismail, and W. Wallace, "Finding hidden groups in communication networks," *submitted to Journal of the Intelligence Community Research and Development (JICRD)*, 2004.

[10] A. Camptepe, M. Goldberg, M. Magdon-Ismail, and M. Krishnamoorthy, "Detecting conversing groups of chatters: A model, algorithms and tests," *Proceedings of IADIS International Conference, Applied Computing 2005*, pp. 145–157, 2005.

[11] J. Baumes, M. Goldberg, and M. Magdon-Ismail, "Efficient identification of overlapping communities," *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 27–36, May, 19-20 2005.

[12] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding comminities by clustering a graph into overlapping subgraphs," *Proceedings of IADIS International Conference, Applied Computing 2005*, pp. 97–104, 2005.