Discovery, Analysis and Monitoring of Hidden Social Networks and Their Evolution

Mark Goldberg, Mykola Hayvanovych, Apirak Hoonlor, Stephen Kelley,
Malik Magdon-Ismail, Konstantin Mertsalov, Boleslaw Szymanski and William Wallace
Rensselaer Polytechnic Institute, Troy, NY, 12180
{goldberg,hayvam,hoonla,kelles,magdon,mertsk2,szymansk}@cs.rpi.edu, wallaw@rpi.edu

Abstract—*Social networks that arise spontaneously and evolve over time have become an important component of ever growing global societies used for spreading ideas and indoctrinating people. Their loose membership and dynamics make them difficult to observe and monitor. We present a set of tools for discovery, analysis and monitoring evolution of hidden social groups on the internet and in cyberspace. Two complementary kinds of tools form a core of our approach. One is based on statistical analysis of communication network without considering communication content. The other focuses on communication content and analyzes recursive patterns arising in it.*

First, we present a software system SIGHTS (Statistical Identification of Groups Hidden in Time and Space), designed for the discovery, analysis, and knowledge visualization of social coalition in communication networks by analyzing communication patterns. We discuss how our algorithms extract groups and track their evolution in Enron-email dataset and in Blog data. The goal of SIGHTS is to assist an analyst in identifying relevant information.

A complementary set of tools uses Recursive Data Mining (RDM) to identify frequent patterns in communication content such as email, blog or chat-room sessions. Our approach enables discovery of patterns at varying degrees of abstraction, in a hierarchical fashion, and in language independent way. We use RDM to distinguish among different roles played by communicators in social networks (e.g., distinguishing between leaders and members). Experiments on the Enron dataset, which categorize members into organizational roles demonstrate that use of the RDM dominant patterns improves role detection.

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 is invaluable for analysis of social coalition and their evolution, yet it must be processed before the relevant information can be extracted. 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 groups to which they belong.

Social network analysis (see [1], [2], [3], [4], [5], [6]) 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. We present here a system called *SIGHTS* design for social network analysis of hidden groups. The main objective of SIGHTS is to provide an umbrella for various algorithms for analyzing communication data with the goal of detecting social coalitions, primarily, "hidden" groups. The graphical user interface of SIGHTS contains facilities to help the analyst examining and manipulating the results of the algorithms.

Any social network group has its structure. For instance, in a company, a social network group composes of the presidents, the secretaries, the managers, the employees and etc. Identifying the role of each member in the group is of primary interest to any group analysis. In any form of communication, traces of personal styles can be identified by authorship analysis to automate artists style classification process [7] to discover intrusion detection [8], and to compress the data [9]. Yet, it was not clear if such stylistic difference can be discern between groups of communicators, based on the roles of sender and receiver of messages. To address this problem, we used *Recursive Data Mining* (RDM) for distinguishing the roles of the communicators in a social group. In general, RDM discovers, in a recursive manner, statistically significant patterns in a stream of data. The key properties of the pattern discovery in RDM include: (i) no restriction of the size of gaps between patterns, (ii) recursive mining in which discovered patterns are replaced by the new token and the mining is repeated on the newly created string, (iii) tolerance to imperfect matching.

2. SIGHTS

The three main modules of SIGHTS (see Figure 1) are: Data Collection/Storage, Data Learning and Analysis; and

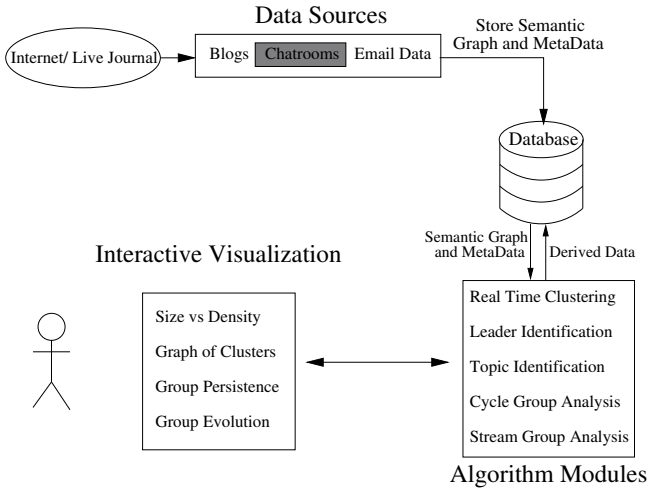

Fig. 1. SIGHTS System Architecture(*currently the link from Chatrooms is not functional*)

Knowledge Extraction/Visualization.

The Data Collection Modules operate 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.

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. Blogs data is collected from LiveJournal.com blogs service provider. 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. 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. This information is also stored in the database and is accessible to other modules of the application.

The Data Processing Modules process the semantic graph and meta data obtained during the data collection to retrieve additional information. They run in parallel with data collection and the identified groups and other data are stored in the database. Using visualization module user may view the data.

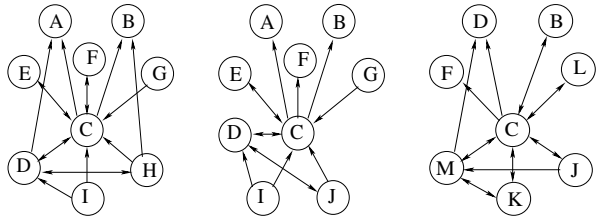

Fig. 2. Evolution of a part of the Enron organizational structure from 2000 - 2002. All three graphs represents frequent structures; the labels indicate actors of the Enron community. Note: that B, C, D, F are present in all three time-intervals.

Temporal group algorithms identify hidden groups in the stream of communications for the user specified time scales [10], [11]. Noteworthy among them are the cycle model algorithms that identify all groups of users who persistently communicate over a time interval and the stream model algorithm that finds groups based on frequently communicating triples followed by merging correlated triples. Such groups usually communicate over multiple time intervals during the period of interest in a streaming fashion. Our algorithms also give the group structure hierarchy and can be modified to track evolution. An example of the evolution of a group found in the ENRON email data set is shown in the Figure 2.

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 [12], [13]. SIGHTS then provides the user with an intuitive way to interact with the results, as shown in Figure 3.

The Opposition Identification Module 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 them.

The threads of comments that appear on LiveJournal.com are split into conversation between pairs of bloggers. The module employs the Support Vector Machine classifier 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.

3. Interactive Visualizations in SIGHTS

SIGHTS provides a GUI for interactive visualization of the results obtained from data collection and data processing mod-

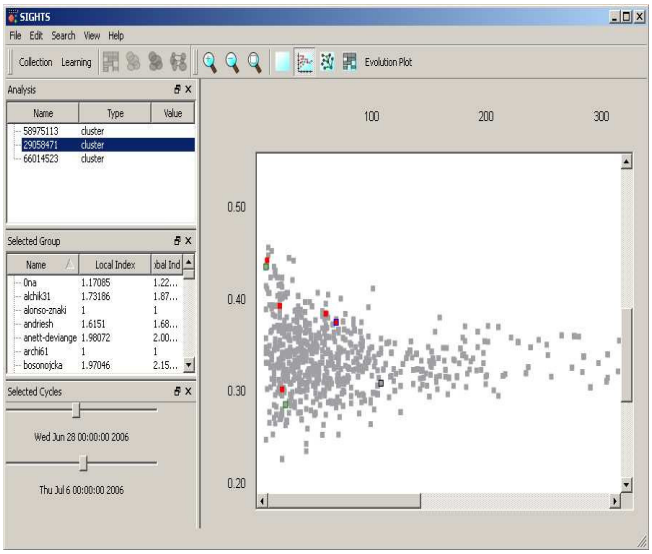

Fig. 3. Size vs. Density Plot of SIGHTS Sample Analysis (*Each dot represents a group, bold dots are groups of high interest level with high amount of communication*)

ules. Currently system supports four different visualizations for the analyst: Size vs. Density plot, graph of Clusters plot, Group Persistence view and Group evolution view. The interactive Graph of Overlapping Clusters 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.

It is also possible to visualize group persistence. Each time cycle has a number of rectangles which belong to this cycle and represent groups. When such group is selected, its members will be shown in the "`Selected Group`" display. The selected group is entirely colored blue while 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.

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. Coalitions that acted 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.

4. Recursive Data Mining

Members of a social group or an organization can usually be partitioned hierarchically into subgroups. The lowest level of such partitioning would contain basic or elementary members, while the highest would identify leader(s). Such hierarchies are amenable for efficient processing via *Recursive Data Mining* (RDM) [14], a new approach for identifying patterns in sequences of tokens. In the RDM process, the recursive (hierarchical) search for repeating patterns captures distinctive features at varying levels of abstraction. Intuitively, at lower levels the patterns obtained are more specific, resulting limiting the potential for valid matches ($\mathcal{M}$). At higher levels, the patterns are more general, increasing the chances for a match. On the other hand, with increasing levels, the number of patterns found decreases monotonically.

In this paper, we present the significant improvement on RDM compared to the previous version [14], using an ensemble of classifiers, each for different level of RDM abstraction, instead of having only one classifier for the entire process. The overall RDM process is outlined in Algorithm 1.

Algorithm 1 Recursive Data Mining

Require: Set of sequences $\mathcal{S}$
Ensure: Sets of patterns (features) $\mathcal{L}$, one for each level
1: $\mathcal{L} = \{\}$
2: $i = 0$
3: $S = \mathcal{S}$ {Level-0}
4: **repeat**
5: $\mathcal{P}_{ALL}$ = *pattern_generation(S)*
6: $\mathcal{P}_{SIG}$ = *sig_patterns(S,* $\mathcal{P}_{ALL}$*)*
7: $\mathcal{D}$ = *get_domi_patterns(S,* $\mathcal{P}_{SIG}$*)*
8: $\mathcal{L}_i = \mathcal{D}$
9: i++
10: $\mathcal{S}$ = *make_next_level(S,* $\mathcal{D}$*)* {Level-i}
11: **until** $\mathcal{M} == \varnothing \vee i == max\_level$
12: **return** $\mathcal{L}$

The process starts with a sliding window of predefined length passing over the input sequence one token at a time. At each stop, patterns with all possible combinations of tokens and gaps are recorded. When pass is completed, the recorded patterns are checked for frequency of their occurrence. Some patterns could be either too specific to a certain text or insignificant because they contain very commonly occurring words. In either case, they are ineffective in classifying the mined text while adding to the computational cost of the algorithm. The "usefulness" of a pattern is computed via a statistical significance test. A pattern is deemed significant if its frequency of occurrence (based on a unigram model) is larger than the expected number of occurrence in a random string. Patterns that are deemed insignificant are eliminated from further consideration.

At each position in the sequence, the tokens in the significant patterns are matched against the tokens in the sequence. The

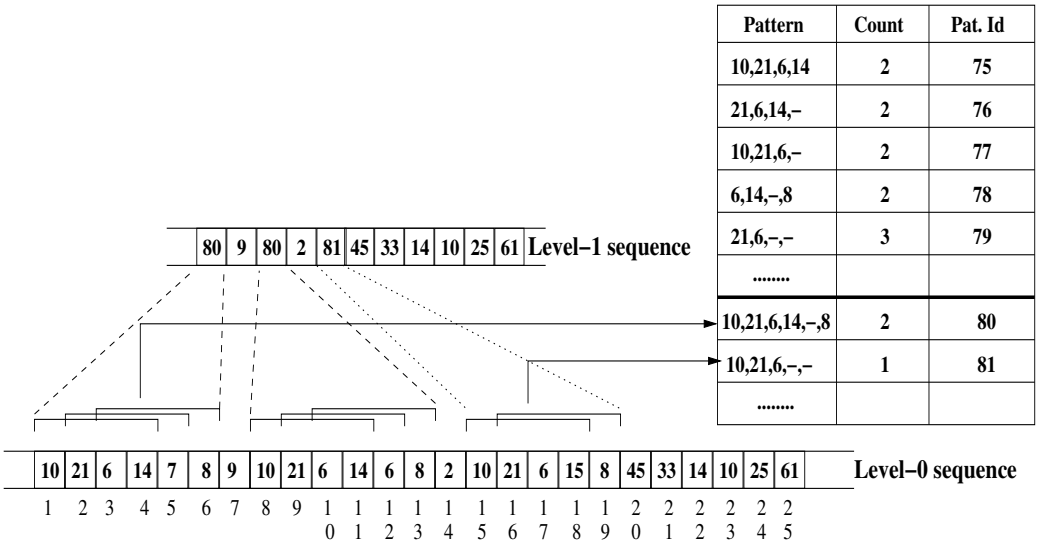

Fig. 4. Sequence Re-writing Step

pattern that has the highest matching score starting at location j in the input sequence is termed as the *dominant pattern* starting at position j. The term dominant pattern is coined from the fact that this pattern dominates over all other significant patterns for this position in the sequence. The matching score is calculated as follows:

$$score(\mathcal{P}i, S_v j) = \begin{cases} 1 & \text{if } \mathcal{P}i = S_v j \\ \epsilon & \text{if } \mathcal{P}i = GAP, \epsilon < 1 \\ 0 & \mathcal{P}i \neq S_v j \end{cases}$$

where $\mathcal{P}i$ is the i^{th} token of the pattern and j is an index over sequence S. ϵ is intended to capture the notion that a match with a gap is not as good as an exact match but much better than a mismatch. Dominant patterns at consecutive positions over the sequence can be merged to form longer dominant patterns. A new token is assigned to each dominant pattern.

During the second pass of the sequence at level v, the sequence for level $v+1$ is formed. The sequence corresponding to a dominant pattern is replaced by the new token for this dominant pattern. Unmatched tokens are copied from sequence S_v to the new sequence S_{v+1}. Figure 4 illustrates this step as well as the merging process.

The training phase uses the dominant patterns generated at each level to construct an ensemble of classifiers, one for each level. The classifiers can be created from any machine learning method, such as Naïve Bayes or Support Vector Machine. Given a set of input sequences, along with the class labels, dominant patterns are generated for each label starting at level 0 to level *max_level*. The union of all dominant patterns at a level v across all input sequences forms the set of feature for classifier at level v. For the ensemble of classifiers, the final posterior class probability is the weighted sum of the class probabilities of individual classifiers. Each classifier is assigned a weight that reflects the confidence of the classifier.

The original input sequences are further split into a new training set and a tuning set. Each classifier in the ensemble trains its model based on the new training set. The confidence value of classifier at level v, $conf(\mathcal{C}_v)$, is defined as the relative accuracy on the tuning set.

The testing phase follows the training phase in terms of the level by level operating strategy. Initially, the frequency of level 0 dominant patterns is counted over the level 0 test sequence. This vector of frequencies forms the feature vector at level 0. Once the level v is done, the next level sequence is generated following the rewriting rules for dominant patterns. This process continues till all levels of dominant patterns are exhausted. Each classifier in the ensemble classifies the test data and the final probability of class C given a sequence x is assigned based on the following weighing scheme

$$\mathbf{P}(C \mid x) = \sum_{i=1}^{max\_levels} conf(\mathcal{C}_i) \times \mathbf{P}_{\mathcal{C}_i}(C \mid x)$$

where x is a test sequence and $\mathbf{P}_{\mathcal{C}_i}(C \mid x)$ is the posterior probability assigned by classifier $\mathcal{C}_i$.

5. RDM Experiments and Results

We compare performance of RDM with performances of classifiers based on, respectively Naïve Bayes, Support Vector Machines and *Predictive Association Rules (CPAR)* [15] that combines the advantages of associative and traditional rule-based classifiers. Support Vector Machines based classifiers performed well on text classification tasks [16]. Since RDM does not use any semantic tools (part-of-speech tagging or synonym groups) in extracting patterns, we compare it with techniques that also do not utilize domain or semantic knowledge.

Our experiments were done on the March 2, 2004 version of Enron dataset, distributed by William Cohen, http://www.cs.cmu.edu/~enron/. The dataset was cleaned to eliminate attachments, quoted text and tables from the body of the email messages and header fields from the email. No efforts were made to correct spelling errors or to expand abbreviations in attempt to reduce the noise.

For our purpose of identifying roles, employees were partitioned into groups based on their organizational role in Enron, as suggested in [17]. Only the roles *CEO, Manager, Trader and Vice-president* were used in our experiments thanks to a large number of employees playing these roles. Since we are concerned with identifying roles based on the sent messages we only deal with the messages in the *Sent* folder of each participant. For each of these roles, the emails are divided into two sets as summarized in Table I.

TABLE I
Dataset for Role Identification

	Training Set	Testing Set	Total	# Sent folders
CEO	1010	250	1260	3
Manager	1403	349	1752	4
Trader	654	162	816	4
VP	1316	327	1643	4
Total	4383	1088	5471	15

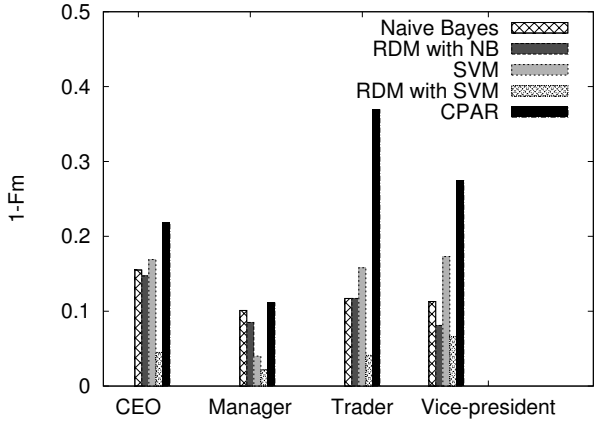

Fig. 5. Binary Classification – F-measure

We used SVMLight to implement SVM and *IlliMine* package to implement CPAR. RDM was used both with Naïve Bayes, and SVM in ensemble of classifiers. In the binary test, given a test message m, the task is to answer *"Is message* m *sent by a person with role r"?* where $r \in \mathcal{R} = \{CEO, Manager, Trader, Vice-president\}$. To compare performance of the classifiers, we used *F-measure* that is the harmonic mean of precision and recall, so higher *F-measure*, better the performance. The performance for the compared classifiers is shown in Figure 5, where the values of 1 - *F-measure* are presented to highlight the differences

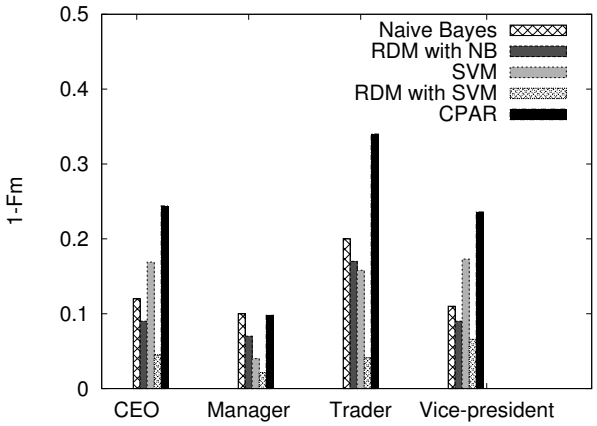

Fig. 6. Multi-class Classification – F-measure

in performances. In terms of the *F-measure*, RDM with NB and SVM performed better than or as good as NB and SVM respectively, for all cases. RDM is better than CPAR under all settings.

The second set of results compares the performance under the multi-class classification test, where the task is to answer *"Which is the most likely role, out of roles* $R_1, \ldots, R_n$*, for sender of message* m*?"* For NB and RDM with NB, the training data is split into 4 groups and probabilities computed for each of the roles. For SVM and RDM with SVM, four sets of datasets are generated, one each for role $[r]$-$[\mathcal{R} \backslash r]$ pairs. The results are shown in figure 6.

To further investigate the results obtained for the multi-class scenario, we performed the *paired t-test* for statistical significance. The paired t-test provides a hypothesis test of the difference between population means for a pair of random samples whose differences are approximately normally distributed. The null hypothesis H_0 states that the difference in performance between RDM and the other methods is not significant. The alternate hypothesis states otherwise. A 20-fold cross validation was performed on the data. The accuracy results obtained therein are used for the t-test. The results are shown in table II. Based on the p-value in Table II we reject the null hypothesis, indicating a definite improvement with RDM. The confidence interval for the mean difference shows that the improvement lies between 1.8% and 3% as compared to NB whereas as compared to SVM (and CPAR) it lies between 8% and 10%.

TABLE II
Results of paired t-test

Classifier Pair	p-value	95% confidence interval
NB vs RDM	1.23E-08	(0.0186 - 0.0292)
SVM vs RDM	1.94E-14	(0.0818 - 0.0984)
CPAR vs RDM	3.74E-13	(0.0821 - 0.1045)

6. Conclusion

We present here set of tools based on SIGHTS and RDM. SIGHTS discovers and analyzes hidden social groups on the internet and in cyberspace and then monitors their evolution. It includes the user friendly visualization tool. Examples of SIGHTS application to social groups evident in Enron-email and Blog data are given. A complementary tool, RDM provides a general framework for identifying hierarchical patterns in a sequence of tokens. RDM was applied to role recognition from the Enron dataset email.

Currently, we are integrating our algorithms with BlackBook-2 [18] which is a software system that integrates the data sources and enables interactive execution of its algorithms. It also offers extended visualization capabilities which we plan to leverage and enhance.

Acknowledgment

This work was partially supported by the ONR Contract N00014-06-1-0466 and also by the U.S. National Science Foundation (NSF) under Grants IIS-0634875, IIS-0621303, IIS-0522672, ITR 0324947 and by the U.S. Department of Homeland Security (DHS) through the Center for Dynamic Data Analysis for Homeland Security administered through ONR grant number N00014-07-1-0150 to Rutgers University. The content of this paper does not necessarily reflect the position or policy of the U.S. Government, no official endorsement should be inferred or implied.

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, **45**(2): 167–256, 2003.

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

[3] J. Sinai, "Combating terrorism insurgency resolution software," *Proc. IEEE Int. Conf. Intelligence and Security Informatics (ISI-2006)*, pp. 401–406.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2006, pp. 44–54.

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

[6] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2003, pp. 137–146.

[7] T. Li and M. Ogihara, "Music artist style identification by semi-supervised learning from both lyrics and content," *Proc. 12th Annual ACM Int. Conf. Multimedia*, 2004.

[8] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," *Proc. 7th USENIX Security Symposium*, 1998, pp. 79–93.

[9] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," *Proc. IEEE INFOCOM*, 2003.

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

[11] A. Camptepe, M. Goldberg, M. Magdon-Ismail, and M. Krishnamoorthy, "Detecting conversing groups of chatters: a model, algorithms and tests," *Proc. IADIS Int. Conf. Applied Computing 2005*, pp. 145–157.

[12] J. Baumes, M. Goldberg, and M. Magdon-Ismail, "Efficient identification of overlapping communities," *Proc. IEEE Int. Conf. Intelligence and Security Informatics*, 2005, pp. 27–36.

[13] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding comminities by clustering a graph into overlapping subgraphs," *Proc. IADIS Int. Conf. Applied Computing 2005*, pp. 97–104.

[14] B. Szymanski and Y. Zhang, "Recursive data mining for masquerade detection and author identification," *Proc. 5th IEEE System, Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004, pp. 424–431.

[15] X. Yin and J. Han, "Cpar: Classification based on predictive association rules," *Proc. SIAM Int. Conf. Data Mining*, 2003.

[16] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," *Proc. 10th European Conf. Machine Learning*, 1998, pp. 137–142.

[17] see http://rabasrv.jhuapl.edu/karma/index.php/Main_Page

[18] see http://rabasrv.jhuapl.edu/karma/index.php/Main_Page (a password is required to access the page).