Egocentric Information Abstraction for Heterogeneous Social Networks

Cheng-Te Li and Shou-De Lin
Graduate Institute of Networking and Multimedia
National Taiwan University
{r96944015, sdlin}@csie.ntu.edu.tw

Abstract

Social network is a powerful data structure that allows the depiction of relationship information between entities. However, real-world social networks are sometimes too complex for human to pursue further analysis. In this work, an unsupervised mechanism is proposed for egocentric information abstraction in heterogeneous social networks. To achieve this goal, we propose a vector space representation for heterogeneous social networks to identify linear combination of relations as features and compute statistical dependencies as feature values. Then we design several abstraction criteria to distill representative and important information to construct the abstracted graphs for visualization. The evaluations conducted on a real world movie dataset and an artificial crime dataset demonstrate that the abstractions can indeed retain important information and facilitate more accurate and efficient human analysis.

1. Introduction

"Information abstraction" generally refers to the summarization and reorganization of the overwhelmed, raw information to a better-accessible representation while still retain the important and meaningful messages. In this study we propose to apply the idea of information abstraction to social networks. Furthermore, a real-world social network can easily contain millions of individuals and relations, and consequently users might not be interested in viewing the network as a whole; rather they prefer learning information of certain instances they are interested in. Hence in this work we would like to tackle the *egocentric* abstraction problem which tries to summarize the information of a given node. Borrowing from social network literatures [13], the node of interests can be referred as the *ego*. The ego node and its directly or indirectly connected neighbors compose a so-called egocentric network, while the information to be retained or discarded depends significantly on a given ego node. As will be shown in the evaluation, an egocentric abstraction can assist human in answering questions like "*what is special about a given entity in the network.*"

One important characteristic of this study is that we pay special attention to the *heterogeneous social networks* (HSN) [13]. A heterogeneous social network contains a set of typed nodes (e.g. nodes can be movies, actors, or directors in the movie domain) and typed edges as relations (e.g. friends, family, directs). Our goal is to perform the egocentric information abstraction in an HSN.

Despite many efforts have been put on social network analysis recently, most existing works assume only one type of nodes and one type of relations in the network. Such network is defined as a homogeneous social network. For example, the Web is regarded as a homogeneous network due to its single node type (webpage) and single relation type (hyperlink). However, for real-world tasks, the heterogeneous social networks provide a much more powerful representation potential since it describes complex relationships among numerous different objects. For example, a movie network shown in Figure 1 takes movies(M), directors(D), writers(W), and actors(A) as nodes and the relations such as <D, *direct*, M>, <M, *hasActor*, A> as tuples, in which the first entry in the tuple is the type of the source node, the second one is the type of the relation, and the third is the type of the target node.

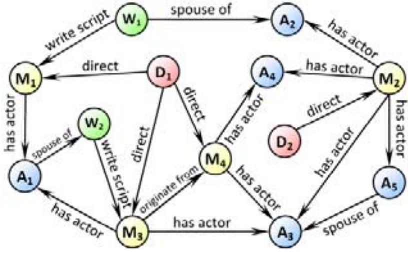


Figure 1. A HSN for movie domain.

The concept of information abstraction has not yet been formally defined in heterogeneous social networks. Though the essences of several works are related to abstraction in some sense, they all suffer a main deficiency for ignoring high-order relationship information. For example, centralities [4] and PageRank [2] aim at finding important nodes in a graph. However, they simply treat any network as a homogeneous one as ignoring node types and relation labels. The same problem occurs in network statistics analysis [13] and community detection [4][8][14] for social networks.

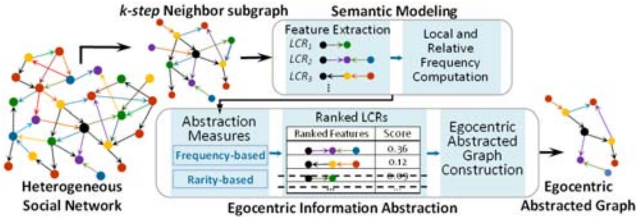

k-step Neighbor subgraph
Semantic Modeling
Feature Extraction
LCR_1
LCR_2
LCR_3
Local and Relative Frequency Computation
Abstraction Measures
Frequency-based
Rarity-based
Ranked LCRs
Ranked Features
Score
0.36
0.12
0.09
Egocentric Abstracted Graph Construction
Heterogeneous Social Network
Egocentric Information Abstraction
Egocentric Abstracted Graph

Figure 2. Flowchart of proposed egocentric abstraction.

To handle the above issues and provide an intuitive, unsupervised, and efficient mechanism for egocentric information abstraction, we propose a model integrating both symbolic and statistic retrieval techniques. The flowchart is shown in Figure 2. We model the semantics of the ego using the surrounding substructure of the *k*-step neighbor subgraph. Besides, three abstraction views, namely local frequency, local rarity and relative frequency are proposed to serve as the distilling criteria for abstraction. Finally, the abstracted graph is constructed for visualization using the distilled information. Our contributions and advantages are listed as follows:

978-0-7695-3689-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ASONAM.2009.38

(1) We define a problem and propose the solution of finding meaningful egocentric abstraction for heterogeneous social networks, which we believe is useful for social network analysis and visualization.
(2) Both topological and relational (i.e. semantics) information are simultaneously considered in our model.
(3) Three abstraction views are introduced, and each of which encompasses its own physical meaning.

This paper is organized as follows. The methodology is described in Section 2 and section 3 reports the experiments. Section 4 describes the related works. We discuss some relevant issues in Section 5 and conclude in Section 6.

2. Methodology

We first provide a formal definition on the egocentric information abstraction problem in HSN:

Given: *(a) a heterogeneous social network H, (b) the query vertex x representing the ego, and (c) the information filtering threshold δ* ($0 \leq \delta \leq 1$).

Outputs: *an egocentric abstracted network of x. The abstraction is a subgraph of H and corresponds to one of the three abstraction measures as will be described in 2.3.*

Definition 1 (Heterogeneous Social Network). *A heterogeneous social network H(V, E, L) is a directed labeled graph, where V is a finite set of nodes, L is a finite set of labels, and* $E \subseteq V \times L \times V$ *is a finite set of edges. An edge is represented by a triple containing the start vertex, the label of the relation, and the end vertex in order. The function types(V)* $\rightarrow \{\{t_1,\ldots,t_j\},\ t_i \in L,\ j \geq 1\}$ *maps each vertex onto its type.*

A heterogeneous social network consists of the topological part and relational part. The nodes are various types of actors, each of which is surrounded by certain combinations of diverse links and nodes. Here we propose to summarize the semantics of a given ego node via combining its surrounding linear substructure together with the statistical dependency measures obtained through certain sampling techniques.

The egocentric information abstraction contains four main stages. First, a set of features are automatically selected and extracted according to surrounding network substructure. They will serve as the basis of summarization. Second, the statistic dependency measures between the features and the ego node are generated. Third, we apply certain distilling criteria to remove less relevant information. Finally, an egocentric abstracted graph can be constructed in an incremental manner that allows the users to visualize the results. The elaboration of these four stages is provided in section 2.1 to 2.4.

2.1 Feature Extraction

We first extract the *k*-step neighbor subgraph $H_{k,x}$ of the ego node *x*. Constraining on the size of the neighborhood is reasonable since it is usually assumed farer away nodes do not have as significant inference as closer ones do. Then we propose to extract the *linear combination of relations* (LCR) as the base to represent the surrounding structure of an ego node. An LCR is defined as an ordered sequence of relations. For example, by taking *k*=2, the set of distinct LCRs of node A_1 in Figure 1 is shown in Table 1. Each LCR can be regarded as a kind of behavior of A_1. Note the inverse edge set E^{-1} is the set of all edges (v_1,l^{-1},v_2) such that $(v_2,l,v_1) \in E$.

Table 1. Two-steps LCRs from A_1 of Figure 1.

LCR_1	*<hasActor^{-1}, writeScript^{-1}>*
LCR_2	*<hasActor^{-1}, direct^{-1}>*
LCR_3	*<spouseOf, writeScript>*
LCR_4	*<hasActor^{-1}, hasActor>*
LCR_5	*<hasActor^{-1}, originateFrom>*

2.2 Statistic Dependency Measures

We design two random experiments based on the LCRs. A random experiment is a trial that can be repeated numerous times under the same conditions, and each outcome is independent and identically-distribution (I.I.D.). In the first random experiment (RE_1), a node *x* is randomly selected from the network, then an edge e_1 starting from *x*, denoted by $<x,e_1,y>$, is randomly selected; further another edge e_2 starting from *y*, denoted by $<y,e_2,z>$, is randomly selected, and the selection goes on until the number of edges chosen reaches *k*. The second one (RE_2) looks very similar to the first one, except that we start from a randomly chosen edge $<a,e,b>$ instead of a node. Next another edge starting from node *b* is chosen, and so on so forth until *k* edges are chosen. The outcomes of either experiment is a path, and based on which we can define two random variables *X* and *L*. *X* represents the starting node of this path and *L* represents the LCR of this path. We use X_1 and X_2 to denote the starting node produced by RE_1 and RE_2, and the same for L_1 and L_2.

With these four random variables, we then define two conditional probability mass functions: $P(L_l=l|X_l=x)$ and $P(X_2=x|L_2=l)$. We call the former *local frequency* of the ego node *x*, which essentially stands for the probability that the LCR of a randomly selected path from *x* is *l*. On the contrary, we call the latter *relative frequency* of an ego node, represents the probability that an ego *x* is involved as the starting node in a given LCR *l*. The former is called "local" because this particular LCR is compared with other LCRs starting from the same ego node (regardless how it distributes in the rest of the network). The latter is called "relative" or "global" since its value depends on how it is distributed in the whole network.

Table 2. Conditional probabilities of RE_1: $P(L_1|X_1)$.

	LCR_1	LCR_2	LCR_3	LCR_4	LCR_5	LCR_6	LCR_7
x_1	0.02	0.08	0	0	0.1	0.3	0.5
x_2	0.3	0.03	0.4	0.25	0	0	0.02
...	…	…	…	…	…	…	…
x_{100}	0	0	0.01	0.07	0.9	0	0.02

After sampling both RE_1 and RE_2 for sufficient amount of times, it is possible to create two tables (e.g. probability values in Table 2 and 3, assuming there are only 7 LCRs) which consist of the corresponding conditional probabilities. We call such tables the *vector-based summarization of nodes*. That is, each row vector in the table is a summarization of one node in the network. Note that in Table 3 we also show the rank (i.e. comparing with all nodes of the same type) of each $P(X_2|L_2)$ below its value inside the parentheses. Besides, the probability of each row sums to 1 in Table 2 while in Table 3 the probability of each column sums to 1.

Table 3. Conditional probabilities of RE_2: $P(X_2|L_2)$.

	LCR_1	LCR_2	LCR_3	LCR_4	LCR_5	LCR_6	LCR_7
x_1	0.05 (76)	0.15 (5)	0.31 (2)	0 (99)	0.06 (88)	0.28 (3)	0.1 (34)
x_2	0.15 (22)	0 (66)	0 (72)	0.7 (1)	0.09 (32)	0.01 (68)	0.08 (21)
…	…	…	…	…	…	…	…
x_{100}	0 (82)	0.01 (60)	0.56 (1)	0.05 (38)	0 (93)	0.02 (51)	0.12 (12)

2.3 Information Distilling

We propose both frequency-based and rarity-based policies to distill different kinds of information for abstraction. Rare and frequent basically occupy two opposite ends of the spectrum. We feel that each reveals either important or potentially interesting information about a given node. Frequent behavior is generally important for pattern recognition and rare events (i.e. those are not supposed to happen but truly happened) sometimes can lead to unexpected discovery. Integrating these two policies with two views (i.e. local and relative view), it is possible to create four kinds of abstraction measures. Here we would like to discuss each of them except the relative rarity measure, which we believe is not as meaningful as the others.

(1) **Local Frequency.** It chooses the frequent $P(L_1|x)$ LCRs of a given ego node x as the important ones. For example, if the threshold δ is set to 0.25, only the top two frequent LCRs in Table 2 (i.e. LCR_6 and LCR_7) would be picked to represent x_I. In other words, LCR_1 to LCR_5 are filtered out since they do not occur as frequent as other LCRs with respect to x. The intuition behind this view is that x is summarized by the frequent behaviors it involves in.

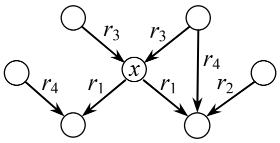
(2) **Local Rarity.** Opposite to (1), the rarity view of abstraction keeps the rare events that happen to x and ignores the frequent ones. For x_I if δ is set to 0.09, LCR_1 and LCR_2 will be distilled while the rest will be ruled out. Here the "rare events" stand for those happen at least once; therefore exclude LCRs with zero conditional probability such as LCR_3. The intuition behind this view is that the rare LCRs could indicate something that should not happen but in fact happens, and thus demands more attention. The other reason such view of abstraction should exist is that the rare events in a large network are generally harder to be detected by human beings than the frequent ones.

(3) **Relative Frequency.** It Table 3, $P(X_2=x|L_2=l)$ in fact represents how frequent the ego x is involved in l compared to other nodes. Since $\Sigma_X P(X_2|L_2)=1$, we can treat each column in Table 3 as a relative comparison among all nodes for a certain LCR l. This measure chooses relatively high $P(X_2=x|L_2=l)$ to represent x. Furthermore, since a heterogeneous social network generally contains different types of nodes, it makes more sense to compare only nodes of the same type when determining the rank of $P(X_2|L_2)$. For instance, it might not be as proper to compare the number of publications among people from different research areas. In the example, LCR_3 and LCR_6 have higher chance to be chosen to represent x_1 since they are ranked relatively high (i.e. 2nd and 3th) for x_1. The intuition is that it picks the features that can best characterize x.

2.4 Abstracted Graph Construction

Until now, our system is capable of generating a condensed feature representation as the abstraction of a given ego node. The system can simply report the distilled LCRs together with the corresponding probabilities to the users. Though it seems to be a reasonable way to present the outputs since $P(L_1|X_1)$ or $P(X_2|L_2)$ can serve as a term that explains why such abstraction is made, an alternative and more understandable presentation is to convert the distilled information back to an HSN. To achieve such goal, we reverse the process of obtaining LCRs to reconstruct a subgraph composed of only the distilled LCRs and the corresponding nodes.

Figure 3 and 4 illustrates this idea. Assume we want to keep the top two scored LCRs and filter out the rest. LCR_1 is first used to match the original network to obtain a subgraph that originates from the ego x and contains all the nodes and edges involved in LCR_1 (see Figure 4(a)). Then the same action is performed for LCR_2. The final egocentric abstraction of x is shown as Figure 4(b).



id	Ranked LCRs	Score
LCR_1	$<r_1, r_4^{-1}>$	0.36 (2)
LCR_2	$<r_1, r_2^{-1}>$	0.18 (5)
LCR_3	$<r_3^{-1}>$	0.09 (10)
LCR_4	$<r_3^{-1}, r_4>$	0.02 (79)
LCR_5	$<r_1>$	0.005 (99)

Figure 3. An example $H_{k,x}$ with the ranked LCRs.

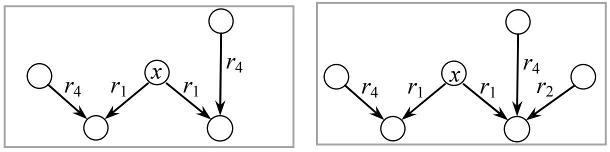


Figure 4. (a) The abstracted graph after adding LCR_1 (b) The final graph after LCR_1 and LCR_2 are added

Note that it is not feasible to produce the abstracted graph by removing the discarded LCRs from the k-step neighbor subgraph since edges involved in one LCR might also occur in others. Therefore eliminating one of them can accidentally eradicate other LCRs.

3. Evaluation

We perform two experiments. The first focuses on demonstrating how the proposed framework can be performed on a real-world movie network by showing the resulting abstracted graph based on different abstraction measures. The second experiment is designed to assess the quality of the abstraction through human studies on a crime dataset. The goal is to find out whether the egocentric abstraction can improve the accuracy and efficiency of human decisions.

3.1 Case Study for a Movie Network

We apply our egocentric information abstraction on a movie dataset to exhibit the abstracted graphs via different abstraction views. The UCI KDD movie dataset [5] is used to construct the HSN containing about 24,000 nodes (9,097 movies, 3,233 directors, 10,917 actors, and some other movie-related persons such as producers and writers) and 126,926 relations. There are 44 different relation types which can be divided into three groups: relations between people (e.g. spouse and mentor), between movies (e.g. remake), and between a person and a movie (e.g. director and actor), which makes it very difficult for human to analyze.

Here we use the node "Meg Ryan", a famous actress, as the ego node to demonstrate the egocentric abstracted graphs. We have to point out that this UCI KDD dataset is incomplete where some information is missing. Therefore certain statistics collected based on it might not reflect the real-world situation. The 2-step neighbor subgraph of "Meg Ryan" is shown in Figure 5. This is not a trivial network analysis since there are 116 nodes, 137 edges and 18 different LCRs.

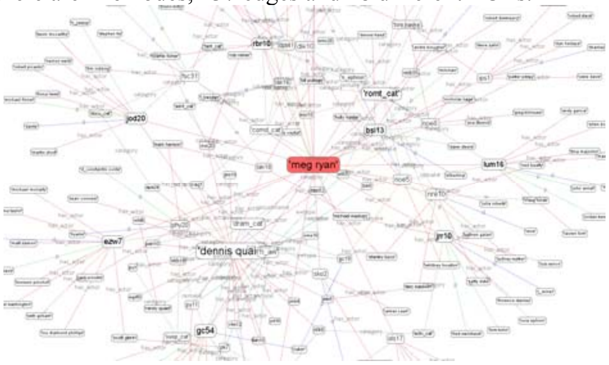


Figure 5. The 2-neighborhood graph of "Meg Ryan."

The abstracted graph of local frequency is shown in Figure 6, which captures the *regular behavior* of Meg Ryan. The filtering threshold δ= 20% is applied in our abstraction (which implies we only keep 20% of the LCRs). We can observe that she played in many movies, especially in comedic, dramatic, and romantic categories. Besides, her husband, Dennis Quaid, is also an actor of many movies. They co-starred in three of them. Such information is not as trivial to obtain from the original graph.

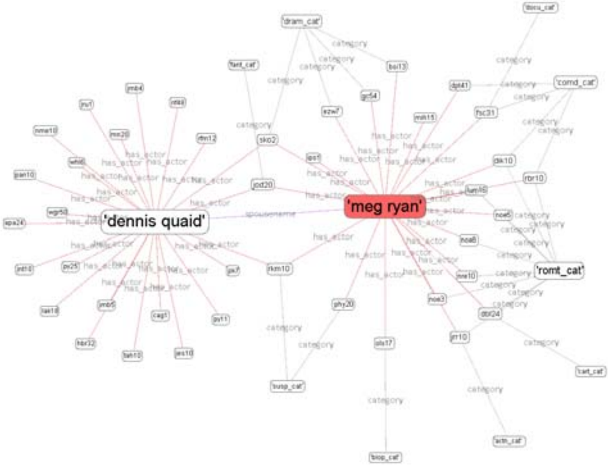


Figure 6. Local frequency of "Meg Ryan."

The local rarity view is shown in Figure 7. It captures the rare behavior of Ryan. We can observe she is also a producer of a movie (i.e. lak16). Besides, her husband's brother (i.e. Randy Quaid) also works in the movie industry (note that only movie-related persons are listed in this dataset). Finally there is a movie she acted (i.e., noe3) whose cinematographer (denoted as 'c' here) is listed in this dataset. This becomes a rare pattern for her since none of her other movies has such information recorded.

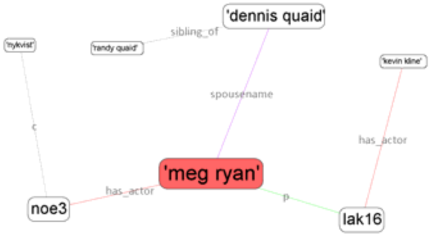


Figure 7. Local rarity of "Meg Ryan."

The abstracted graph of relative frequency is shown in Figure 8, which compares the behavior of Meg Ryan with other *actors* (note: not all other persons in the dataset) and identifies the behavior she significantly involves in. We can observe an interesting behavior of her as she acted in relatively large amount of remade movies comparing with others. Also she produced a movie (i.e., lak16) and such behavior does not appear to be frequently among other actors. Finally, one path of her based on local rarity measure, namely his husband's sibling is also a movie person, turns out to be rare among other actors as well, and thus becomes a relatively frequent behavior of her (that is, there are very few others in this dataset whose husband's sibling is also a movie person).

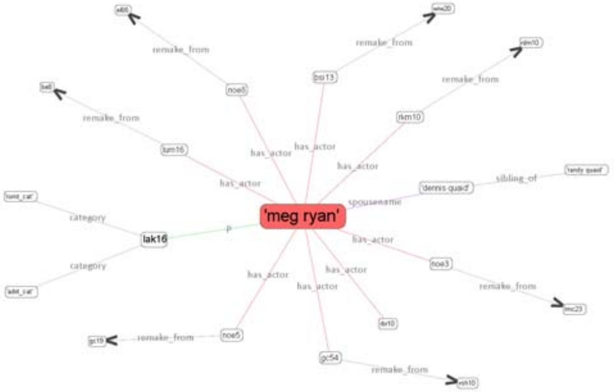


Figure 8. Relative frequency of "Meg Ryan."

In this case study, we have used a heterogeneous movie network to demonstrate which kinds of information can be revealed through which egocentric view. We have also demonstrated that through our abstraction mechanism, it is possible to discover not only some expected details (e.g. Ryan acted in many romantic movies) but also some unexpected yet potentially interesting facts (e.g. Ryan acted in many remade movies and produced a movie) about the ego node. It might even satisfy some hard-core fans by revealing certain information about her ex-husband.

3.2 Human Study for Crime Identification

In this experiment, we evaluate our abstractions through a study of the quality of human decision-making. The goal of this evaluation is three-fold: 1) to know whether and which of the abstracted networks can assist human subjects to make more accurate decisions. 2) To see whether the abstractions can reduce the time needed to make a decision. 3) To learn whether the abstraction can improve human subjects' confidence about their decisions.

The dataset we used is a simulated crime dataset developed during US Defense Advanced Research Projects Agency's Evidence Extraction and Link Discovery Program [9] for evaluating link discovery algorithms like group detectors, pattern matchers, and etc. The data is generated by a simulator of Russian organized crime (i.e. Mafiya) that simulates the process of ordering, planning, and executing criminal activities such as murders or gang wars with many possible variations and records an incomplete and noisy picture of these activities in the files (e.g. financial transaction, phone call, email, somebody being observed at a location, somebody being killed by someone unknown, etc.). It has about 9000 nodes and twice as many edges with 16 node types of objects (e.g. bankAccount, Mafiya, and industry) and 31 different relation types (e.g. perpetrator and victim). There are 42 gang nodes and 20 contract murder events. Besides, it is noisy since some relations are missed or labeled incorrectly, which could cause difficulties for analysts.

The experiment setup is as follows. For each dataset we first choose 10 plausible gang nodes, among which three are truly involved in the high-level crime events (i.e. involving in a gang war or trying to take over an industry). Then for each gang node, the original *k*-step neighbor subgraph (we set *k*=3 here) together with three kinds of abstracted graphs (δ=20%) are generated. We present these four kinds of HSNs separately to 20 users, and ask them to select three (out of ten) nodes that they believe are most likely to commit high-level crimes. In this sense, we can examine how many out of the 20*3=60 possible outcomes were picked correctly. To avoid interference among different tasks, the IDs of all candidate instances are randomly given for each task and subjects are not instructed in which order they should pursue. Before the experiment, the subjects were asked to study the background knowledge of the crime domain so they understand the meaning of each type of node and relation.

The results are displayed in Table 4. We also show the improvement over *k*-step neighbor subgraph in the first column and 95% confidence interval for average time and confidence.

Table 4. The results with 95% confidence interval.

	Avg. Precision	Avg. Time (minutes)	Avg. Confidence (1~5, 5 is the highest)
k-step Neighbor Subraph	39/60	36.6 ± 6.6	3.15 ± 0.36
Local Frequency	41/60 (+3.3%)	18.9 ± 5.9	3.20 ± 0.35
Local Rarity	44/60 (+8.7%)	13.9 ± 3.7	3.45 ± 0.33
Relative Frequency	47/60 (+13.3%)	10.9 ± 2.2	3.73 ± 0.39

In terms of accuracy, the results show that users can usually perform better (the improvement can be as high as 13.3%) while using the abstracted networks comparing with the original one. Our explanation is that although certain information is lost after abstraction, it is likely the critical messages are remained while some noise is filtered out, which leads to better results. The major improvement, as shown in the second column of Table 4, lies in efficiency. Users spend significantly less amount of time (<50%) to reach better results. The improving on accuracy, efficiency, and confidence demonstrates that the abstraction is capable of facilitating better human analysis.

In this dataset, there are some *key evidences* that can indicate the high-level events. After analyzing the abstracted graphs manually, we have realized that each abstraction view more or less captures different parts of those key evidences. For example, a kind of LCR that represents "the gang has hired some middleman intending to pursue something illegal" happens only to the high-level crime participants; therefore it can be highlighted using the relative frequency view, which becomes an important evidence for the human subjects to make the right decision. This could be the major reason that this view eventually leads to the best results among others.

4. Related Works

Graph Summarization. Graph summarization mainly aims at generating compact and understandable representation for a large graph. L. Zou et al. [16] propose the *summarization graph* using the topological information of the original homogeneous graph to handle the subgraph search problem. S. Navlakha et al. [7] use the principle of Minimum-Description-Length to summarize single-relational graphs. They allow lossless and lossy graph compressions with bound on the indicated error to produce the *aggregate graph*. However, it is not clear how these approaches can be adopted to HSNs. Y. Tian et al. [12] introduce the OLAP-style operations to summarize multi-relational graphs, where users can apply drill-down and roll-up to control summarized resolutions. However, this work utilizes only one-step links or nodes and has not provided the egocentric view.

Visual Analysis for Network Abstraction. Network visualization aims at efficiently displaying a large network by drawing the structural data with some simple analyses for human explorations. P. Appan et al. [1] summarize key activity patterns of social networks in the temporal domain using a ring-based design. L. Singh et al. [11] develop visual mining program to help people understand the entire multi-mode networks at different abstraction levels. Z. Shen et al. [10] divide abstraction to structural and semantic parts, and presented a visual analytics tool, OntoVis, where the relations in heterogeneous social networks were reduced based on the concept of network ontology. Again, they consider only links in one step neighborhood, where we exploit higher-order relational information with statistical sampling to identify important information.

Mining in Heterogeneous Networks. While most existing social network related works concentrate on homogeneous networks, some efforts are gradually shifted to heterogeneous networks recently. D. Cai et al. [3] address the community detection problem in heterogeneous networks through learning an optimal linear combination of user-given relations.

J. Zhang et al. [15] do recommendations via heterogeneous Web network by a modified random walk with a pair-wise learning algorithm. Lin et al. [6] propose some unsupervised mechanisms to identify abnormal instances from HSNs.

5. Discussions

There are several issues worthy of further discussions:

a) *The efficiency*. To estimate the probabilities accurately, we need to sample a sufficient amount of paths, which becomes the bottleneck of our approach. However, a technique called likelihood weighting, which has been applied successfully in the inference procedure of Bayesian Networks, can be applied to force the occurrence of some rare events. Then the likelihood can be reweighted based on the frequency of the forced decisions.

b) *Parameters*. There are two parameters to control the level of abstraction: the k in k-neighborhood and δ as the trimming threshold. Each of them has its own physical meaning. Increasing k can enlarge the size (or radius) of the network and increasing δ can boost the density of the graph. Therefore we recommend determining k based on the number of nodes and links in the network, and adjusting δ based on the number of different link types.

c) *Union or Intersect measures*. In reality there can be more than three measures of abstraction since views can be integrated. For example, one can union local frequency and local rarity measures to visualize both frequent patterns and rare events in the abstraction. One can also intersect the local frequency and relative frequency views to make sure only behavior that is both frequent and representative are shown.

6. Conclusions

In this paper we present a method for egocentric information abstraction for heterogeneous social networks. We believe it can be applied to creating a node-based search engine for social networks as well as realizing social network visualization. Here we provide an alternative view about our approach. An intuitive approach to graph abstraction is to identify certain seems-to-be irrelevant edges and vertexes to remove. However, it is non-trivial how such removal can be made (either manually or automatically) when the information is represented as a heterogeneous social network where nodes and edges are mixed together to form complicate patterns. To answer this challenge, we argue that the abstraction should be pursued in a *retaining* manner rather than an *eliminating* manner. That is, we should build the abstracted graph by trying to retain important or relevant information instead of removing the irrelevant ones. Therefore in this paper we propose a two-level abstraction schema. The first level of abstraction is to transform the original network representation into a vector-space representation using symbolic modeling and sampling techniques. The reason to perform such transformation is that now we are allowed to pursue the second-level abstraction as applying some simple and intuitive criteria to determine which portion of the information should be retained. Finally our goal can be achieved through incrementally transforming the retained vectors back to the original domain of networks.

7. References

[1] P. Appan, H. Sundaram and B. L. Tseng. Summarization and Visualization of Communication Patterns in a Large-Scale Social Network, In *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'06)*, 371–379, 2006.

[2] S. Brin and L. Page. The Anatomy of Large-scale Hypertextual Web Search Engine. In *Proc. of Intl. World Wide Web Conference (WWW'98)*, 107–117, 1998.

[3] D. Cai, Z. Shao, X. He, X. Yan and J. Han. Mining Hidden Community in Heterogeneous Social Networks. In *Proc. of ACM SIGKDD Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD'05)*, 58–65, 2005.

[4] D. Chakrabarti and C. Faloutsos. Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Survey*, 38(1), 2006.

[5] S. Hettich and S. D. Bay. The UCI KDD Archive. http://kdd.ics.uci.edu, University of California, Irvine, Department of Information and Computer Science, 1999.

[6] S. D. Lin and H Chalupsky. Discovering and Explaining Nodes in Semantic Graph. *IEEE Transactions on Knowledge and Data Engineering*, 20(8), 1039–1052, 2008.

[7] S. Navlakha, R. Rastogi and N. Shrivastava. Graph Summarization with Bounded Error. In *Proc. of ACM SIGMOD Intl. Conference on Management of Data (SIGMOD'08)*, 419–432, 2008.

[8] M. E. J. Newman and M. Girvan. Finding and Evaluating Community Structure in Networks. *Physics Review*, 2004.

[9] R. Schrag. A Performance Evaluation Laboratory for Automated Threat Detection Technologies. In *Proc. of Performance Measures of Intelligent System Workshop* (PerMIS'06), 2006.

[10] Z. Shen, K. L. Ma and T. Eliassi-Rad. Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 12(6), 1427–1439, 2006.

[11] L. Singh, M. Beard, L. Getoor and M. B. Blake. Visual Mining of Multi-Modal Social Networks at Different Abstraction Levels. In *Proc. of Intl. Conference on Information Visualization* (IV'07), 672–679, 2007.

[12] Y. Tian, R. A. Hankins and J. M. Patel. Efficient Aggregation for Graph Summarization. In *Proc. of ACM SIGMOD Intl. Conference on Management of Data (SIGMOD'08)*, 567–580, 2008.

[13] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, UK, 1994.

[14] X. Xu, N. Yuruk, Z. Feng and T. A. J. Schweiger. SCAN: A Structural Clustering Algorithm for Networks. In *Proc. of ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining (KDD'07)*, 824–833, 2007.

[15] J. Zhang, J. Tang, B. Liang, Z. Yang, S. Wang, J. Zuo and J. Li. Recommendation over a Heterogeneous Social Network. In *Proc. of Intl. Conference on Web-Age Information Management (WIAM'08)*, 309–316, 2008.

[16] L. Zou, L. Chen, H. Zhang, Y. Li and Q. Lou. Summarization Graph Indexing: Beyond Frequent Structure-Based Approach. In *Proc. of Intl. Conference on Database Systems for Advanced Applications*, 141–155, 2008.