Egocentric Information Abstraction and Visualization for Heterogeneous Social Networks

Cheng-Te Li and Shou-De Lin
Computer Science and Information Engineering Department
Graduate Institute of Networking and Multimedia
National Taiwan University
Taipei 106, Taiwan
{r96944015, sdlin}@csie.ntu.edu.tw

ABSTRACT
Social Network is a powerful representation and visualization schema that allows the depiction of the relationships information between entities. However, for real-world tasks, the constructed heterogeneous networks are usually too complex for users to perform advanced investigations. In this paper, an unsupervised mechanism is proposed for egocentric information abstraction and visualization in heterogeneous social networks. Our abstraction consists of two levels. The first level of abstraction is a summarization process that maps the egocentric heterogeneous network onto a vector-space domain by identifying linear combination of link types as features and computing several statistical dependencies as feature values. The second level of abstraction focuses on using four diverse abstraction criteria to distill representative and/or informative messages, and use them to reconstruct the abstracted networks for visualization. The evaluations were conducted on a real world movie dataset and an artificial crime dataset. The experimental results not only demonstrate the abstracted networks but also show that such abstraction and visualization can facilitate more accurate and efficient crime investigation for human subjects.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *Data Mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information Filtering*; J.4 [**Computer Applications**]: Social and Behavior Sciences – *Sociology*.

General Terms
Algorithm, Human Factors, Measurement.

Keywords
Social networks, heterogeneous networks, egocentric, information abstraction, visualization.

1. INTRODUCTION
"Information abstraction" generally refers to the summarization of a raw, overwhelmed information into a more concise form while still retain the important and meaningful message. The significant information is retained and possibly reorganized to a human-understandable representation while the trivial one is filtered and discarded. Our work explores the possibility of applying the concept of information abstraction to the social network or graph domain. Furthermore, to facilitate advanced mining or information retrieval on a social network, we argue that such abstraction has to emphasize on an *egocentric* view. Borrowing from social network literatures [19], the node of interests can be referred as the "*ego*". The ego node and its directly or indirectly connected neighbors compose a so-called ego-centered or egocentric network. The egocentric information abstraction highlights on the micro viewpoint of the network. In other words, the kind of information to be retained or discarded should depend on the ego node that users intend to pay attention to. Therefore, as can be seen in our experiments, an egocentric abstraction can assist human in answering questions like "which individual might be suspicious", or "what is special about a specified movie star".

WWW 2009, April 20-24, 2009, Madrid, Spain.

One important characteristic of this study is that we pay special attention to a kind of data structure called *heterogeneous social networks* [19]. A heterogeneous social network contains a set of typed nodes (i.e. 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 ego-centered information abstraction in this kind of heterogeneous social network.

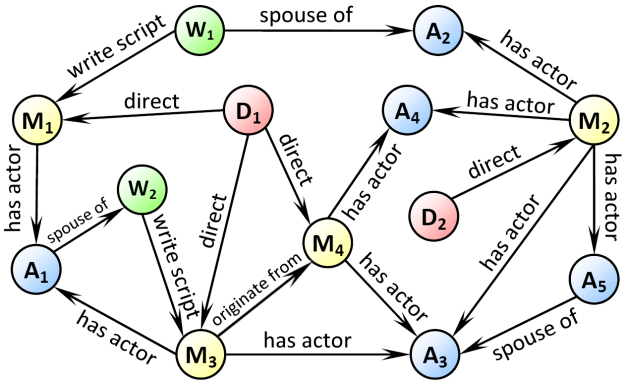


Figure 1: A heterogeneous network for movie domain. The capital letter of each node stands for its type: M(movie), D(director), A(actor), and W(writer).

Despite many efforts have been put on social network analysis in recent years, most of the existing methodologies assume that there are only one object type on nodes and one relation type on edges in the network, which is defined as *homogeneous* social networks. For example, a Web can be regarded as a homogeneous social network considering there is only one type of node (webpage) and relation (hyperlink). However, in the real-world modeling, the heterogeneous social networks do provide a much 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 corresponding

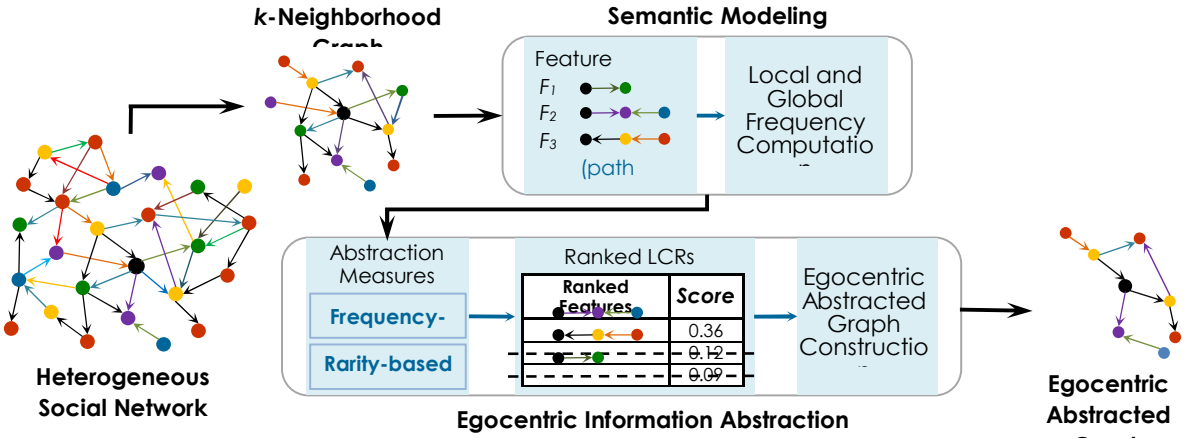


Figure 2: The Flowchart of Proposed Egocentric Information Abstraction in Heterogeneous Social Networks.

relationships as tuples such as "<D, *direct,* M>", "<M, *has_actor,* A>", "<M, *originate_from,* M>", "<A, *spouse_of,* A>", and "<W, *write_script,* M>", where the first element in the tuple represents the type of the source node, the second element as the type of the relations, and the third element as the type of the target node.

The concept of "information abstraction" has not yet been formally defined in the domain of heterogeneous social network. Nevertheless, the essences of several research topics in social network analysis such as centrality analysis, group detection, etc., are indeed related to information abstraction in some sense. Despite this, they all suffer certain deficiencies and our major goal is to design an information abstraction mechanism for networks that deals with those problems. Below we discuss several main deficiencies of the existing ideas in social network abstraction:

- *Losing topological knowledge.* Degree distribution, network diameter, average path length, and other network statistics are global features which exploit simple statistics to summarize certain characteristics of a network [5][12]. They can be regarded as a kind of abstraction for social networks. Although such abstractions can be attained easily, they suffer a major drawback as not keeping sufficient topological information of a social network.
- *Ignoring higher-order relationship information*. There are works aiming at taking the network structure or topology into consideration for analysis, such as PageRank [3] and Centrality analysis. However, those methods simply treat any network as a homogeneous one by ignoring the difference between node types and relation labels. The same problem occurs in network statistics and community detection. One of the major contributions of our framework lies in the unbiased, fully automatic mechanism that takes beyond single-step relation information into account during the process of abstraction. That is, our approach not only considers the types of relation but also model the correlation amount relations as well as the dependency between set of relations and entities.
- *Non-egocentric view of the world.* Existing community detection frameworks such as cohesive subgroup finding [13][21] tries to identify representative groups in a network. While providing a macro view on how the social network can be simplified as a whole through identifying groups, such mechanism suffers a drawback of not being able to produce an egocentric analysis about a specific entity. Our work focuses on generating a kind of abstraction whose focus may shift depending on which ego is chosen.
- *User-unfriendly outputs*. Existing information abstraction research rarely considers the issue of presenting their results in a user-friendly form. For example, some graph mining methods such as frequent graph pattern mining [18][22] can also be regarded as a kind of non-egocentric abstraction. However, those works did not explicitly discuss how the results can be displayed for comprehensible and concise purpose. Here we argue that a suitable social network abstraction mechanism should accompany with a comprehensible visualization means. Since social networks are inherently visual, we hope that our abstraction model can be seamlessly fitted into a visualization mechanism. The existing visualization research, however, has not really considered the issue of abstraction in a deeper sense. For instance, L. Freeman [6] designed several principles for network visualization by taking color, position, shape, and size into considerations; other works [1][7] focus on developing intelligent visualization techniques that facilitates better explorations. Our work, to a certain extent, tries to bridge the gap between mining and visualization on social networks.

To conquer the above drawbacks and provide an unsupervised (which implies that human biases can be minimized), intuitive, and efficient mechanism for egocentric information extraction and visualization, we propose a model that integrates both symbolic and statistic information retrieval techniques. Our framework first tries to model the semantics of any given ego node, and then to distill the representative and/or informative features. The semantics of any ego node is modeled by its surrounding substructure (i.e. within k steps from the ego node) together with the label information. Furthermore, four abstraction views, namely local frequency, local rarity, relative frequency and relative rarity are proposed to serve as the distilling criteria for abstraction. Finally, it tries to construct the abstracted graph for visualization using only the distilled information. The flowchart of our system is shown in Figure 2.

The contributions of this work can be summarized as:

- We have proposed a research problem as finding abstraction for heterogeneous social networks, which aims at a kind of egocentric analysis which facilitates further visualization of data. Our design has the following advantage
  a) The topological (or structure) and relational (or semantics) information are simultaneously taken into account for abstraction purpose.
  b) We provide four views of abstraction, each of which encompasses its own physical meaning.
  c) The abstracted results can be represented as the simplified heterogeneous social network for visualization.
- We have implemented such egocentric abstraction system and conducted experiments on both real-world and synthetic dataset. The experiments not only demonstrate the usability of our approach but also show that the designed egocentric abstraction can assist human analyst in making more accurate, efficient, and confident decision.

The rest of the paper is organized as follows. The model and methodology of analysis are discussed in the next section. The experiment results are reported in Section 3. Section 4 describes the related works and section 5 discusses some relevant issues. We conclude in the final section.

2. Methodology

We first provide a formal definition on the egocentric information abstraction problem in heterogeneous social networks:

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

Outputs: *visualized egocentric abstracted of x, each of which belongs to the subgraph of H and corresponds to one of the four proposed abstraction views.*

Definition 1. (Heterogeneous Networks) *A heterogeneous 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. Given a triple representing an edge, the source, label, and target map it onto its start vertex, label, and edge vertex, respectively. The function types(V)* $\rightarrow$ $\{\{t_1,\ldots,t_j\},\ t_i \in L,\ j \geq 1\}$ *maps each vertex onto its set of type labels.*

A heterogeneous network consists of the topological part and relational part. The nodes are various types of actors, each of which is surrounded (up to a certain distance) by certain combinations of diverse links (relations) and nodes. In other words, the semantics of a node in a given heterogeneous network is captured by the information of its neighborhood links and nodes. Hence, we propose to summarize the semantics of a given ego node through combining its surrounding linear substructure (i.e. sequences of labels) together with the statistical dependency measures obtained through certain sampling techniques.

The egocentric information abstraction can be divided into four main stages. First, a set of features are automatically selected and extracted according to surrounding network substructure of the specific node. They will serve as the basis of summarization. Second, the statistic dependency measures between the features and the ego node are generated to represent the ego node. Third, we apply certain distilling criteria to remove less relevant or less informative 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

For egocentric abstraction, we first extract the surrounding subgraph of the ego up to a given length k (in our experiments, we choose k=2 or 3). Constraining on the size of the neighborhood is reasonable since in network analysis it is usually assumed farer away nodes do not have as significant inference as closer ones. Based on the given range k, a *k-neighborhood graph* H_{ki} of the given node i is extracted. H_{ki}, contains all the nodes and their corresponding relations within k steps to the ego. For example, the 2-neighborhood graph H_{2A1} of A_1 in Figure 1 is illustrated in Figure 3.

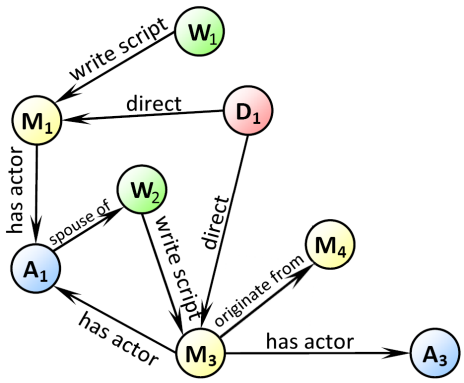


Figure 3: 2-neighborhood graph H_2 of A_1 in Figure 1 (k=2).

The next step is to select a set of representative features as the basis to summarize the ego node. We first exemplify the idea using Figure 3, starting by listing two-step (k=2) paths that start from a specific node, A_1, as shown in Table 1. Note that the inverse edge set E^{-1} is the set of all edge (v_1, l^{-1}, v_2) such that $(v_2, l, v_1) \in E$. Besides, a path p in H is a sequence of edges $(e_1,e_2,\ldots,e_n)$, $n \geq 1$, such that each $e_i \in E$ and $target(e_i) = source(e_{i+1})$.

Table 1: Two-step paths from A_1 of Figure 3.

$path_1$	$A_1 - hasActor^{-1} - M_1 - writeScript^{-1} - W_1$
$path_2$	$A_1 - hasActor^{-1} - M_3 - writeScript^{-1} - W_2$
$path_3$	$A_1 - hasActor^{-1} - M_1 - direct^{-1} - D_1$
$path_4$	$A_1 - hasActor^{-1} - M_3 - direct^{-1} - D_1$
$path_5$	$A_1 - spouseOf - W_2 - writeScript - M_3$
$path_6$	$A_1 - hasActor^{-1} - M_3 - hasActor - A_3$
$Path_7$	$A_1 - hasActor^{-1} - M_3 - originateFrom - M_4$

Table 2: Two-steps LCRs from A_1 of Figure 3.

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

We propose to use a *linear combination of relations* (*LCR*) as the base to represent the surrounding structure of a given node.

For example, the paths in Table 1 can further be condensed to a set of LCR as shown in Table 2. Each linear combination of relation can be regarded as a kind of behavior of A_1.

2.2 Statistic Dependency Measure

Given the LCRs are generated as the bases, we then design two random experiments applying to the k-neighborhood network. A random experiment, by definition, is an experiment, trial, or observation that can be repeated numerous times under the same conditions, and each outcome is I.I.D. In the first random experiment (RE1), we randomly select a node (say, *x*) from the network, then we randomly select an edge starting from *x* (say, <*x*,L1,*y*>), further we randomly select another edge starting from *y* (say, <*y*,L2,z>), so on and so forth. We stop when the number of links chosen is k, where k corresponds to the predefined k-neighborhood. The second random experiment (RE2) looks very similar to the first, except that we start from a randomly chosen edge (say, <*a*,R,*b*>) instead of a node. Next we randomly pick another edge starting from b. Again, this goes on until k links are chosen. The outcome for 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 (e.g. in this example, one realization of *X* is *x*) and L represents the LCR of this path (e.g. in this example, one instance of L is <L1,L2…L*k*>). We would use X1 and X2 to denote the starting node generated by RE1 and RE2, respectively, and same applies to L1 and L2.

Table 3: Conditional probabilities of RE1: P(L|X).

Feature / Node	L1	L2	L3	L4	L5	…	L100
X1	0.2	0.24	0.11	0	0	…	0.2
X2	0.31	0.01	0.4	0.01	0	…	0.08
…	…	…	…	…	…	…	…
X1000	0	0	0.11	0.02	1	…	0.2

Table 4: Conditional probabilities of RE2: P(X|L).

Feature / Node	L1	L2	L3	L4	L5	…	L100
X1	0.01 (99)	0.04 (155)	0.1 (2)	0.5 (1)	0 (500)	…	0 (300)
X2	0.11 (22)	0.1 (1)	…	…	…	…	0.08 (221)
…	…	…	…	…	…	…	…
X1000	0	0	…	…	…	…	0.12 (12)

With these four random variables, we then define two conditional probability mass functions P(L1=*m*|X1=*n*) and P(X2=*n*|L2=*m*). P(L1=*m*|X1=*n*), which we call local frequency of the ego node, essentially stands for the probability that if from n one randomly picked *k*-step LCR in fact equals *m*. On the contrary, P(X2=*n*|L2=*m*), which we call the relative frequency of an ego node, represents the probability that a specific node *n* is involved as the starting node in an LCR whose form is *m*. The former probability is considered as "local" because this particular LCR feature is compared with other LCR feature starting from the same ego node (regardless how it distributed in the rest of the network). The later conditional probability is called "relative" or "global" frequency measure since this value will depend on how this feature is distributed in the whole network.

After sampling both RE1 and RE2 for sufficient amount of times, it is possible to create two tables, i.e., Table 3 and Table 4, which consist of the estimation of the corresponding conditional probabilities. We call such tables the vector-based summarization of nodes. Note that the probability of each row sums to 1 in Table 3 while in Table 4 the probability of each column sums to 1.

2.3 Information Distilling

We propose two distilling policies, frequency-based and rarity-based policy to distill different kinds of information for abstraction. Rare and frequent basically occupy two opposite ends of the spectrum. We feel that each reveal 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 certain unexpected discovery. Integrating these two policies with two views (i.e. local and relative view), we can create four kinds of abstraction measures (summarized in Table 5), and each serves its unique purpose.

Table 5. Four abstraction measures from different aspects.

	Absolute (local)	**Comparative (global)**
Frequent	(1) *Local Frequency*	(3) *Relative Frequency*
Rare	(2) *Local Rarity*	(4) *Relative Rarity*

Below we provide some intuitive discussion for each of them using an example:

Given Table 3 and Table 4, now it is possible to generate the vector-based summarization of an ego node by identifying one row from each table corresponding to it. Table 6 and Table 7 describe the vector from Table 3: P(L|X) and Table 4: P(X|L) for a given node x respectively (assuming there are only 7 LCR's in this dataset). Note that in Table 7 we also list the ranking of each P(X|L) comparing with all the *same-type nodes* in the network, The values are shown inside the parentheses. For example, in Table 7, P(X=x|L4)=0 ranks 999 implies there are 999 same-type nodes in the whole network since it possesses the smallest probability.

Table 6: The Local-based Vector of X.

	L1	L2	L3	L4	L5	L6	L7
T1:P(L|x)	0.01	0.02	0	0	0.1	0.3	0.5

Table 7: The Relative-based Vector of X

	L1	L2	L3	L4	L5	L6	L7
T2:P(x|L)	0.05 (769)	0.15 (5)	0.11 (2)	0 (999)	0.01 (888)	0.1 (3)	0.1 (34)

(1) *Local frequency*: It basically chooses the frequent P(L|x) elements from the vector as important ones. For example, if the threshold δ is set to 2/7, the system will pick only the top two most frequent LCR (i.e., L6 and L7) to represent x. In other words, L1 to L5 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 most frequent sequential patterns that it involves.

(2) *Local Rarity*: Opposite to (1), the rarity view of abstraction keeps the rare event that happens to x and ignores the frequent ones. In this example given δ=2/7, L1 and L2 will be distilled while the rest will be ruled out. Note that the "rare event" considers only those happens at least once, therefore excludes those whose conditional probability is 0 such as L3 and L4. The intuition behind this view is that rare LCR could indicate something that shouldn't happen but in fact happens, and thus demands more attention. The other reason that such view of abstraction should exist is that rare events in a large network are generally harder to detect manually than frequent ones.

(3) *Relative Frequency*: The view of (3) and (4) utilizes Table 4 instead of Table 3. The conditional probability P(X=x|Lk) in fact represents how frequent the ego node x is involved in a kind of LCR. Since $\sum_X P(X \mid L) = 1$, it is possible to treat each column in Table 4 as a relative comparison among all Xs for a given LCR. This kind of view believes P(X=x|Lk) is representative for x if this value is relatively high comparing with other nodes. Furthermore, since a heterogeneous social network generally contains different types of nodes (e.g. actors, directors, movies, etc), it makes more sense to compare only *nodes of the same type* while determining the rank of P(X|L). For example, it might not make sense to compare the number of publications among people from different research areas. In this example, L3 and L6 will be chosen to represent x since they are relatively high (i.e. ranked 2nd and 9th) comparing with other nodes of the same type. The intuition behind this view is that it chooses the features that can best characterize x.

(4) *Relative Rarity*: Similarly to what (2) is to (1), here we claim that some features that happen relatively rare to x might also indicate something worth reporting. In this example L1 and L5 will be distilled since they do not occur as frequently to x as they do to others. This view basically tells us something that x does but not as frequently as other nodes.

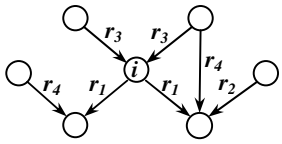
2.4 Abstracted Graph Construction

Until now, our system is capable of generating a condensed feature representation as the abstraction of a given ego node. One plausible output form will be to report the distilled LCR and their conditional probabilities to the users. Although it seems to be a reasonable outputs since the P(X|L) or P(L|X) can serve as a term that explains why such abstraction is made, an alternative and probably more understandable representation will be to translate the distilled information back to a graph. In this section, we would like to propose a method that does so.

Recall that the LCRs of the ego node were obtained through enumerating the plausible combination of relations starting from the ego. We can perform the reverse engineering of such process (can be regarded as a kind of pattern recognition) to generate a subgraph that is composed of only the distilled LCR's and their corresponding nodes.

Figure 4 provides a graphical analysis of such process. Assuming based on relative frequency we decide to keep the top 2 ranked LCRs and filter out the rest. We can first use L1 to match the original network to obtain a subgraph that originates from the ego node and contains all the nodes involved in L1 (see Figure 5(a)). Subsequently we can perform the same action on the second highest LCR until the threshold is reached. In this example, the final abstracted graph of the ego node looks like Figure 5(b).

Note that it is not feasible to create such condensed graph by removing the ruled out LCR from the original network. It is because the links involved in one LCR might also occur in others; therefore eliminating one of them completely will sometimes cause the other LCRs to disappear, which can lead to error when the disappeared LCRs happen to be the representative ones.



id	*Ranked LCRs*	*Score*
L1	i-r_1-*-r_4^{-1}-*	*0.36 (2)*
L2	i-r_1-*-r_2^{-1}-*	*0.08 (5)*
L3	i-r_3^{-1}-*	*0.09 (10)*
L4	i-r_3^{-1}-*-r_4-*	*0.02 (79)*
L5	i-r_1-*	*0.005 (99)*

Figure 4: An example H_k with the corresponding *AFL*.

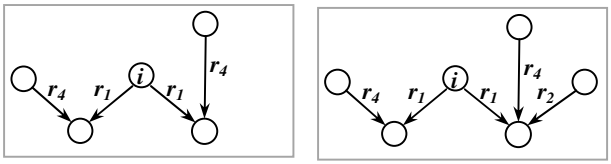


Figure 5: (a) The abstracted graph after adding L1 (b) The final graph after both L1 and L2 are added

3. EVALUATIONS

The evaluations can be divided to two parts. The first experiment focuses on demonstrating how the proposed framework can be performed on a real-world movie network dataset. We would demonstrate the resulted 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 network dataset to exhibit the abstracted graphs according to different abstraction views. The social network is generated from extracting entities and relations from UCI KDD Archive movie dataset [8]. In this network, there are about 24,000 nodes representing movies (9,097), directors (3,233), actors (10,917), and some other movie-related persons (500) such as producers and writers (the numbers in parentheses show the number of different instances for each node type). We also extract 126,926 relations between these nodes. Totally, there are 44 different relation types in the movie network, which can be divided into three groups: relations between people (e.g., spouse and mentor), relations between movies (e.g., remake), and relations between a person and a movie (e.g., director and actor). The amount of diverse relations makes it a complicated heterogeneous social network for human to analyze.

Here we use a "Meg Ryan", a famous actress, as the ego node to demonstrate the egocentric abstracted graphs. We have to first point out that this UCI KDD dataset is not a complete dataset while some information is missing. Therefore certain statistics collected based on it might not reflect the real-world results. The

k-neighborhood graph of "Meg Ryan" is shown in Figure 6, where k=2. Despite the seems-to-be small neighborhood size, from Figure 6 we can learn that it is already very complicated since there are 116 nodes, 137 edges and 18 different LCRs. In the following, the filtering threshold δ is set to around 20%.

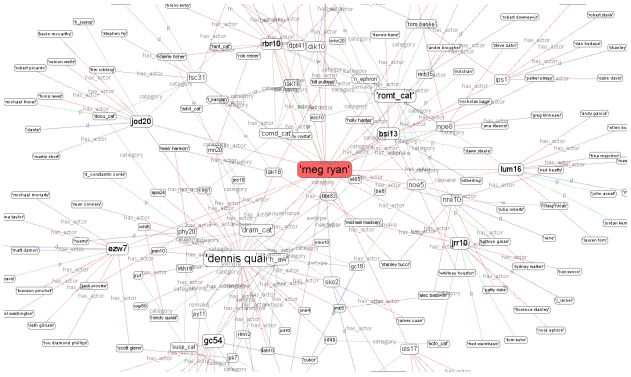


Figure 6: The 2-neighborhood graph of "Meg Ryan."

First of all, the abstracted graph of local frequency is shown in Figure 7, which captures the *regular behaviors* of Meg Ryan. We can observe that she has been acted in many movies, especially for comedic, dramatic, and romantic categories. Besides, her husband, Dennis Quaid, is also an actor that of many movies. They co-starred in three of them.

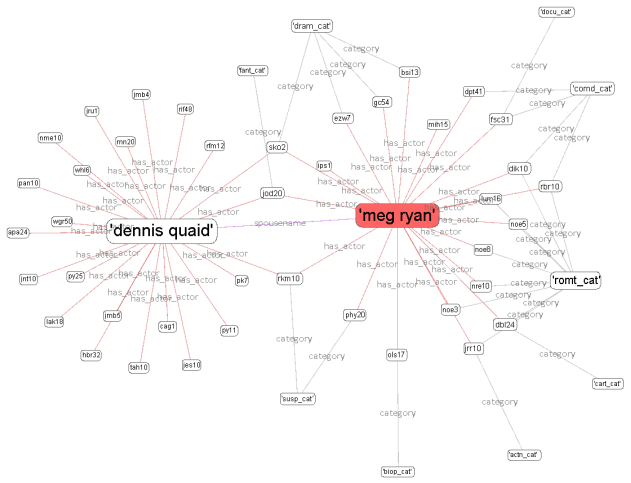


Figure 7: Local frequency of "Meg Ryan."

The abstracted graph of local rarity is shown in Figure 8. This is to capture the rare but existed behaviors of Meg Ryan. We can observe that she is also a producer of a movie (i.e., lak16), which is the only movie she produced according to this dataset. In addition, her husband's brother (named Randy Quaid) also works in movie industry (since only movie-related persons are listed in this dataset). Finally there is a movie she acted (noe3) whose cinematographer (denote as 'c' here) is listed in this dataset. This becomes a rare pattern since for other movies she has played in their cinematographer are not listed.

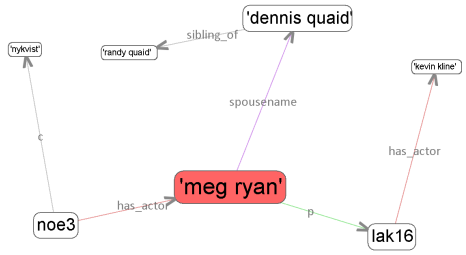


Figure 8: Local rarity of "Meg Ryan."

Third, the abstracted graph of relative frequency is shown in Figure 9, which compares the behaviors of Meg Ryan with other *actors* (note: not all other persons in the dataset) and find those behaviors which is significantly to her. We can observe an interesting behavior of her that she acted in relatively many remade movies. Also she produced a movie (i.e., lak16) and such behavior does not appeal frequently among other actors. Finally, one rare path of her in Figure 8, 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 is very few others in this dataset whose husband's sibling is also a movie person).

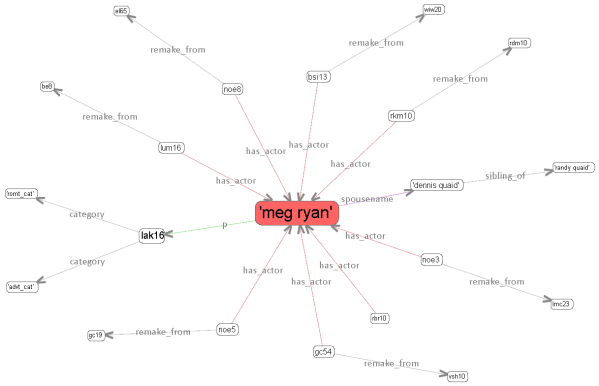


Figure 9: Relative frequency of "Meg Ryan."

Finally, the abstracted graph of relative rarity is shown in Figure 10. This identifies something she did, but not as unique/special as other behaviors of her. We can see that her movies received three awards (i.e., 'cg_aw', 'h_aw', and 're_aw'). Since there are other persons whose movies won awards, this turns out to showcase that she also has similar behavior but not as frequent as some other people. Also it is interesting to know that she played in one movie that was later reproduced to another one.

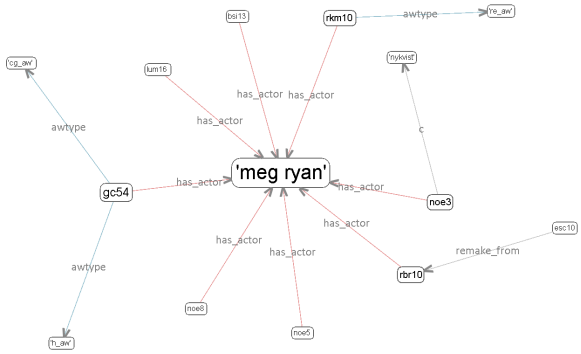


Figure 10: Relative rarity of "Meg Ryan."

In this case study, we have used a heterogeneous movie network to demonstrate which kinds of information can be revealed through which egocentric views. We have also demonstrated that through our abstraction mechanism, it is possible to find not only some expected details (e.g. Ryan acted in many romantic movies) but also some unexpected yet 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: Crime Identification

In the second experiment we evaluated the quality of the abstracted visualization by applying our system to a heterogeneous network in crime domain and ask human subjects to identify the crime participants for us. The goal of the evaluation is three-fold: First, we want to know whether and which of the egocentric abstracted graphs can assist human subjects in making more accurate decisions in terms of identifying the criminal participants. Secondly, whether the proposed abstractions can reduce the time the subjects need to perform such identification. Finally we would like to learn whether the human subjects feel more confident of their decision given the abstracted information.

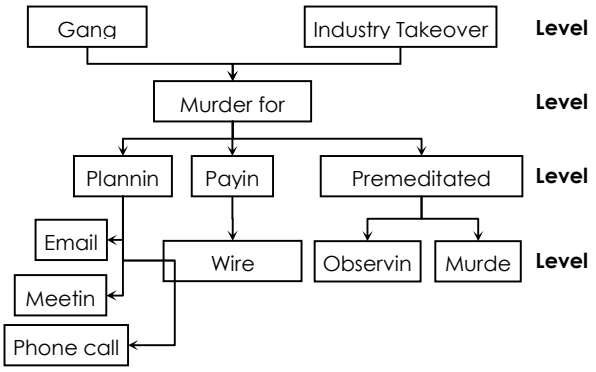


Figure 11: Event-type hierarchy of the simulated Russian organized crime data.

The crime dataset we used is part of a large suite of simulated dataset developed during the US Defense Advanced Research Projects Agency (DARPA)'s Evidence Extraction and Link Discovery Program for the purpose of evaluating link discovery algorithms such as pattern matchers, group detectors, etc., (see [14] for additional contexts). The data was generated by a simulator of a Russian organized crime (or Mafiya) domain that simulates the whole process of ordering, planning, and executing high-level criminal activities such as murders for hire or gang wars with a large number of possible variations and records an incomplete and noisy picture of these activities in the generated evidence files (e.g., financial transaction, phone calls or email, somebody being observed at a location, somebody being killed by someone unknown, etc.). The hierarchy of event types is shown in Figure 11. The highest level events, gang wars and industry takeovers, both involve lower level events such as contrast murders, which in turn involve some planning, financing, execution, etc.

The dataset we employ contain 9,429 nodes, and 16,257 links. There are 16 different node types representing objects and events and 31 different link types representing the relationships between those nodes, as shown in Table 8. It contains 42 Mafiya groups, and 20 contract murder events. On the other hand, the observability of the dataset is quite low, which means some of the events are not shown in the data (the higher level an event is, the higher change it would be omitted, and level 5 events are completely unseen in the data). Besides, the noise of the dataset is occurred to some extent. That is, some information about the links are missed or even labeled incorrectly. Such data can, presumably, cause some problem for the human analyst.

The experiment setup is as follows: we first choose 10 plausible gang nodes among which three were truly involved in the highest level events (gang war and industry takeover). For each gang node, four different views of egocentric abstracted graphs were generated. Together with the original k-neighborhood graph (we choose k=3 in this experiment), we will present five sets of data and each set consists of ten visualized graphs, each ego-centralized on the corresponding candidate. To avoid interference among different tasks, the IDs of all candidate instances are randomly given for each task. Five sets of resulted graphs are shown to a total of 20 users (subjects were not instructed in which order of datasets they should pursue) and the users were asked to select three (out of ten) nodes that are most likely to commit high-level crimes. Therefore we can examine how many out of the 20*3=60 possible outcomes were picked correctly. Before the experiment, the subjects were asked to study the background knowledge of this domain so they understand the meaning of each relation and the node types.

Table 8: Node and relation types for the terrorism network.

Node Types	Relation Types	
BankAccount	accountHolder	orgMiddleman
Business	callerNumber	payee
Email	ceo	payer
Industry	dateOfEvent	perpetrator
Mafiya	deliberateActors	phoneNumber
Meeting	deviceUsed	receiverNumber
Murder	employees	recipient
MurderForHire	eventOccursAt	relatives
Observing	geoSubregions	sender
Paying	hasMember	socialParticipants
Person	mediator	subevents
PhoneCall	hitContractor	transferMoneyFrom
PhoneNumber	hitman	transferMoneyTo
Planning	objectsObserved	victimintended
PremeditatedMurder	operatesInRegion	vor
WireTransfer	orgHitman	

The five generated graphs of one criminal node is illustrated in Figure 12 to 16, which are corresponding to the original 3-neighborhood graph, local frequency, local rarity, relative frequency, and relative rarity in order. Note that the filtering threshold δ is set to 0.2, which implies we only keep 20% of the LCRs during abstraction. And the black nodes are nodes representing criminal candidates.

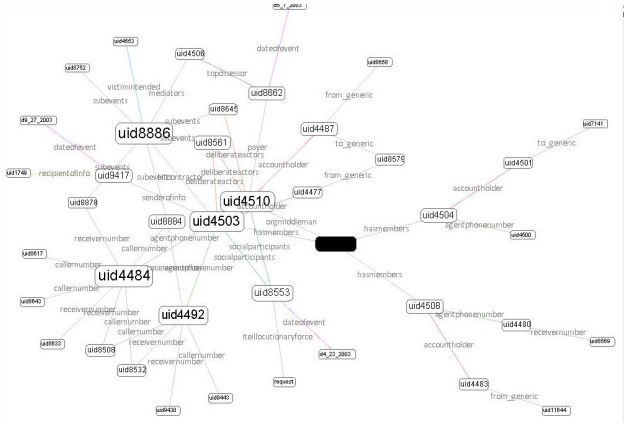


Figure 12: The original 3-neighborhood graph.

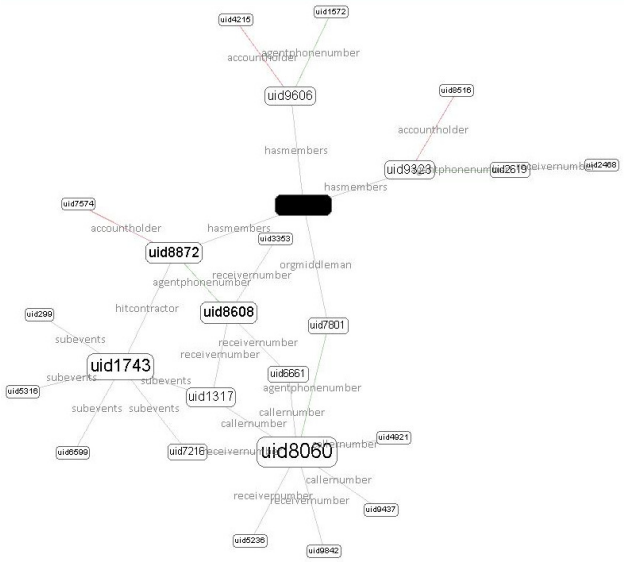


Figure 13: Abstracted graph of local frequency.

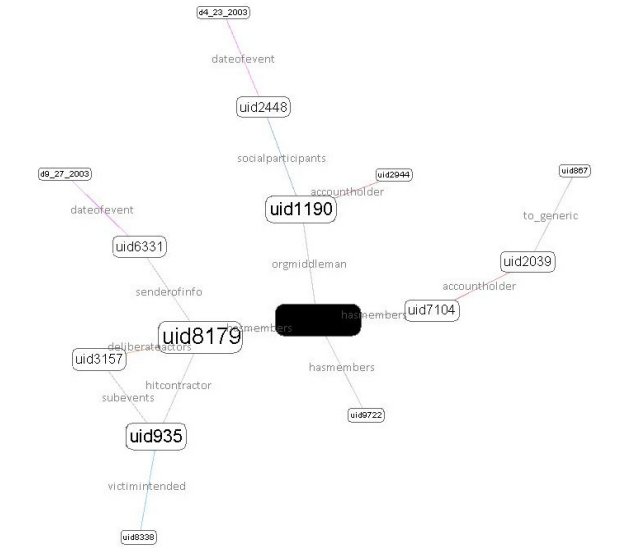


Figure 14: Abstracted graph of local rarity.

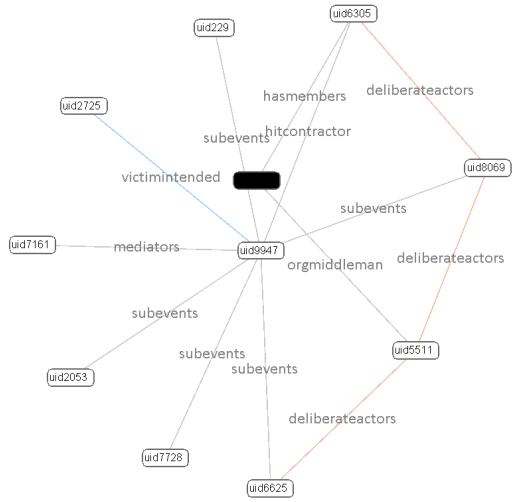


Figure 15: Abstracted graph of relative frequency.

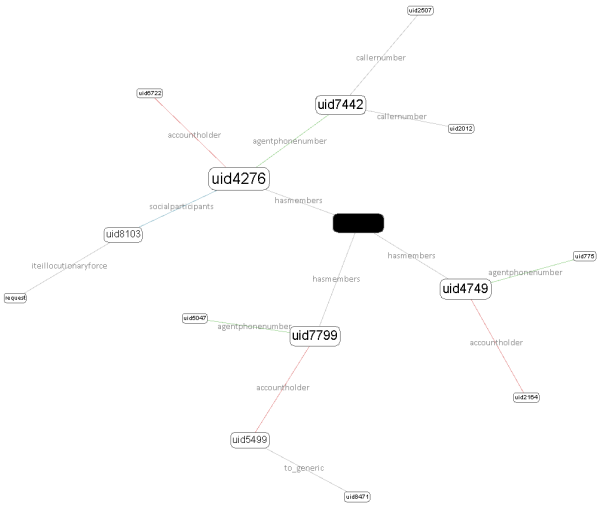


Figure 16: Abstracted graph of relative rarity.

The results are displayed in Table 9. We also show the improvement over k-neighborhood graph in the first column and 95% confidence interval for average time and confidence.

Table 9. Raw *k*-neighbor graph and four abstraction measures from different aspects with their 95% confidence interval.

	Avg. Precision	Avg. Time (minutes)	Avg. Confidence
***k*-Neighborhood Graph**	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
Relative Rarity	40/60 (+1.6%)	11.3 ± 2.5	2.85 ± 0.27

In terms of accuracy, the results show the users can at least do as good as using the original graph while using the abstracted ones. Since the non-abstracted graph contains the complete information, it makes sense to assume that subjects can do as good as using the abstracted ones at the cost of spending more time on the data. Our explanation for the reason that the users can even perform better (the improvement can be as high as 13.3%) in the abstracted graph is that although certain information is lost during abstraction, it is likely the critical messages are remained while *some noise is filtered out*, and therefore lead to better results. The major improvement, as shown in the second column of Table 9, lies in efficiency. The results show that users utilize significantly less amount of time (<50%) to reach at least equal-quality results. The improving on both accuracy and efficiency truly demonstrate that the abstraction is capable of facilitating further human analysis since it retains the critical information and significantly remove uninformative one.

In this dataset, there are some "key evidences" that can indicate the high-level events. After analyzing four kinds of abstracted graphs manually, we have realized 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 is the major reason that this view eventually leads to the best results among others. We have also realized that the relative rarity view does not reveal significant improvement over accuracy and even results in worse confidence. We believe this is because naturally this view reveals the behavior that occurs but not as frequent as others, which might not be as helpful to identify suspicious instances as other kinds of behaviors.

4. RELATED WORKS

Graph Summarization: Graph summarization mainly aims at generating compact and understandable summarized representation for a large graph. It is a relatively new topic and has been tackled from the view of database management recently. L. Zou et al. [24] proposed a Summarization Graph, which captured the topological information of the original homogeneous graph to handle the sub-graph search problem. It is not trivial how their approach can be adopted to a heterogeneous graph. Y. Tian et al. [17] introduced the OLAP-style operations to summarize multi-relational graphs, where users can apply drill-down or roll-up to control the resolution of summarization. However, they did not consider the egocentric view, nor do they take the linear combinations of relations into consideration (only immediate links of nodes are considered). Their summarization also lacks an easily- accessible graphic outputs like our visualization. S. Navlakha et al. [11] proposes a Minimum-Description-Length based principle to summarize for single-relational graph. Their representations of graphs allowed for both lossless and lossy graph compression with bounds on the indicated error, and produced a coarse-level aggregate graph. Nevertheless, it is not clear how their method can be applied to heterogeneous social network, and it only provides the macro view.

Visual Analysis for Network Abstraction. Network Information visualization aims at efficiently displaying a large scale network by drawing the structural data with some simple analyses for human explorations. We have seen three works aiming at integrating the network abstraction into visualization. P. Appan et al. [2] summarized key activity patterns of social networks in the temporal domain through a ring-based visualization design. L. Singh et al. [16] developed visual mining software to help people understand the entire multi-relational networks at different abstraction levels. Z. Shen et al. [15] further divided the abstraction to structural and semantic parts, and presented a visual analytics tool, OntoVis, where the relations in heterogeneous networks were reduced based on the concept of network ontology. However, all of these works suffer from no providing micro view to facilitate user's exploration. Furthermore, unlike our framework who takes multiple link combination and their statistic dependency into account, the above works consider only links in one step neighborhood, therefore cannot fully integrate the topological information with the relational 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. In the early period, W. Xi et al. [20] modeled the Web as a collection of multi-type interrelated data objects. Later D. Cai et al. [4] addresses the community detection problem in heterogeneous networks through learning a optimal linear combination of user-given relations. J. Zhang et al. [23] works on recommendation in a heterogeneous Web social network by a modified random walk along with a pair-wise learning algorithm. Most of these listed works take advantage of the information in the heterogeneous social networks. However, they do not really focus on summarizing the behavior of a certain object in the network. Besides, since they are dealing with domain specific problems, it is not clear how much extra efforts are needed to adapt the proposed methods into different domains. One important advantage of our abstraction lies in that it is domain independent and can be applied to create abstraction in different domains without the need to resorting to domain experts. It not only saves time of identifying training or annotated data but also avoids human biases in the analysis. Lin and Chalupsky have proposed some unsupervised mechanisms for heterogeneous social network analysis. However, the work has been mainly focused on anomaly detection rather than abstraction and visualization [9][10].

5. DISCUASSIONS

There are several issues worthy of further discussions:

a) *The efficiency of our algorithm*: 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. Furthermore, the advantage of exploiting sampling technique for frequency estimation is to facilitate the design of an anytime algorithm. That says we can still produce results of certain quality given insufficient time or resources, and the quality of the results can improve with the increase of the time or resources.

b) *Parameters*: There are two parameters the users can use to control the level of abstraction: the k in k-neighborhood and δ as the trimming threshold and 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 δ based on the number of different link types.

c) *Union or Intersect views*: In reality there can be more than four views of abstraction one can exploit since views can be integrated. For example, one can union local frequency and local rarity views 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 behaviors that are both frequent and representative are shown.

6. CONCLUSIONS

In this paper we present a method for egocentric information abstraction for heterogeneous social networks. Here we provide an alternative view about our approach: Conventionally the process of graph abstraction is regarded as trying different methods to identify certain seems-to-be irrelevant edges and vertexes to eliminate. However, it is non-trivial how such decision can be made (either manually or automatically) when the information is represented as a complex, heterogeneous social network where a node can connect to others through different types of links. To answer this challenge, 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 in this transformed domain we are allowed to apply our second-level abstraction as applying some simple and intuitive filtering 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] E. Adar. GUESS: A Language and Interface for Graph Exploration. In *Proc. of ACM SIGCHI International Conference on Human Factor in Computing Systems (CHI'06)*, 2006.

[2] 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)*, 2006.

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

[4] 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)*, 2005.

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

[6] L. Freeman. Visualizing Social Network. *Journal of Social Structure*, 1(1), 2000.

[7] J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proc. of IEEE Symposium on Information Visualization (InfoVis'05)*, 2005.

[8] 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.

[9] S. D. Lin and H. Chalupsky. Using Unsupervised Link Discovery Methods to Find Interesting Facts and Connections in a Bibliography Dataset. *SIGKDD Explorations*, 5(2), 173–178, 2003.

[10] 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.

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

[12] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45:167–256, 2003.

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

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

[15] 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.

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

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

[18] W. Wang, C. Wang, Y. Zhu, B. Shi, J. Pei, X. Yan and J. Han. GraphMiner: A Structural Pattern-mining System for Large Disk-based Graph Databases and Its Applications. In *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD'05)*, 879–881, 2005.

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

[20] W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan and W. Y. Ma. Link Fusion: A Unified Link Analysis Framework for Multi-Type Interrelated Data Objects. In *Proc. of International World Wide Web Conference (WWW'04)*, 319–327, 2004.

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

[22] X. Yan and J. Han. gSpan: Graph-based Substructure Pattern Mining. In *Proc. of IEEE International Conference on Data Mining (ICDM'02)*, 721–724, 2002.

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

[24] L. Zou, L. Chen, H. Zhang, Y. Li and Q. Lou. Summarization Graph Indexing: Beyond Frequent Structure-Based Approach. In *Proc. of International Conference on Database Systems for Advanced Applications (DASFAA'08)*, 141–155, 2008.