Mining Heterogeneous Social Networks for Egocentric Information Abstraction

Cheng-Te Li and Shou-De Lin

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 combination of relations as features and compute statistical dependencies as feature values. These features, either linear or cyclic, intend to capture the semantic information in the surrounding environment of the ego. Then we design three abstraction measures to distill representative and important information to construct the abstracted graphs for visual presentation. The evaluations conducted on a real world movie dataset and an artificial crime dataset demonstrate that the abstractions can indeed retain significant 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 humanly-understandable representation while still retaining the important and meaningful messages. Figure 1 shows that overwhelming and complex information can usually hinder further manual analysis. In this work, we exploit the idea of information

Cheng-Te Li,
National Taiwan University,
e-mail: d98944005@csie.ntu.edu.tw

Shou-De Lin,
National Taiwan University,
e-mail: sdlin@csie.ntu.edu.tw

abstraction in heterogeneous social networks. Further, given the fact that a real-world social network can contain thousands or even millions of individuals and relations, and therefore users might not be interested in the network as a whole, rather they are particularly interested in the information of certain instances. Therefore, we propose the *egocentric* abstraction problem attempting 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*. The egocentric analysis highlights the micro view of the network. In other words, the information to be retained or discarded depends on the ego that users focus on. Thus, as will be shown in the evaluation, an egocentric abstraction can assist human in answering questions such as "which individual might be suspicious" or "*what is special about the specified movie star*" more efficiently.

Fig. 1: The 2-neighborhood graph of "Meg Ryan."

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, and directs). Our goal is to perform the egocentric information abstraction in an HSN.

Although there are already various successful proposals on social network analysis (SNA), most assume there is only single type of nodes and single type of relations in a network. This kind of social network is defined as homogeneous social networks [13]. For example, both the Web and the citation graph (i.e., nodes are authors and edges represent co-authorship) can also be regarded as a homogeneous social network because there is only one type of node (i.e., webpage or paper) and relation (i.e., hyperlink or citation link). However, in the real-world different types of objects can be connected through different kinds of relationships, therefore it is natural to define different types of entities and relations in a social network. In this sense, a more universal data structure, called heterogeneous social network, has been proposed to describe the complex relationships (i.e., a set of typed edges) among entities. For example, a heterogeneous movie network shown in Figure 2 takes movies (M), directors (D), writers (W), and actors (A) as nodes, and their corresponding relationships as tuples such as $< D_1, direct, M_1 >, < M_1, hasactor, A_1 >, < A_1, produce, M_1 > and < M_3, originatefrom, M_4 >$, where the capital letter in the tuple stands for the type of source node, and the second element stands for the type of relations. Note that in general there could have multiple relationships in between two entities, like the relations of "*has actor*" and "*produce*" between A_1 and M_1.

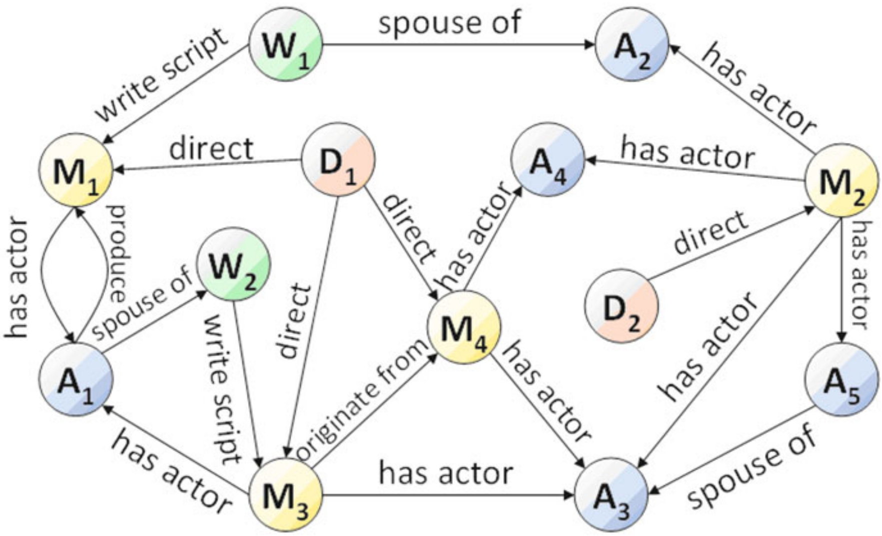


Fig. 2: A heterogeneous social network for movie domain. The capital letter of each node stands for its type: M(movie), D(director), A(actor), and W(writer). Besides, there are five relation types, including *"write script", "has actor", "spouse of", "direct", "produce",* and *"originate from"* in this example.

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 from 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.

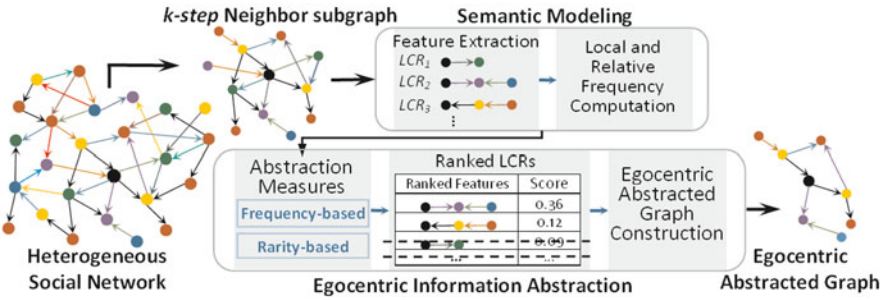


Fig. 3: Flowchart of proposed egocentric abstraction.

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 3. We model the semantic behaviors of the ego using the surrounding substructure of the k-step neighbor subgraph. The Ego-based features (linear combination of relations) are extracted to represent the entities, and the corresponding feature values are calculated using sampling techniques on nodes and paths. Moreover, we propose three abstraction measures, namely local frequency, local rarity and relative frequency, to serve as the distilling criteria to perform abstraction from diverse views. Finally, we construct the abstracted graph for visualization using the distilled information. Our contributions and advantages are listed as follows:

1. We define and propose the solution for discovering representative egocentric abstraction in heterogeneous social networks to facilitate advanced analysis and visualization on social networks.
2. Both topological and relational (i.e., semantic) information are integrated as the linear combination of relations in our model to capture the behaviors of the given ego node. Besides, we propose the linear and cyclic features to describe the ego node.
3. Three abstraction views are introduced, and each of which encompasses its own physical meaning.
4. We conduct experiments on both natural and synthetic datasets to demonstrate the validity and usefulness of our system. The results indicate that our abstraction mechanism can indeed capture certain important information of the ego and provide accurate and efficient solutions for crime identification.

This paper is organized as follows. We describe the related work in Section 2 and the methodology is presented in Section 3. Section 4 reports the experiments. We discuss some relevant issues in Section 5 and conclude in Section 6.

2 Related Works

We divided the existing works related to information abstraction on network data into the following four categories:

Graph Summarization. Graph summarization is about generating the compact summarized representation for a large graph. L. Zou et al. [16] propose summarizing a graph using the topological information of the original homogeneous graph. It is not a trivial matter questioning how their approach can be adopted to heterogeneous graphs. Y. Tian et al. [12] introduce the OLAP-style operations to summarize multi-relational graphs, in which users can apply drill-down and roll-up to control summarized resolutions. However, they only use the immediate links of nodes and the high-order relationship information is ignored. 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 bounds on the indicated error, and produce the aggregate graph. Nevertheless, it is not clear how their method can be applied to a heterogeneous network.

Network Abstraction for Visual Analysis. 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 fashion. L. Singh et al. [11] develop a visual mining program to help people understand the entire multi-mode networks at different abstraction levels, in which the abstraction is performed by merging or dividing among different types of entities. Z. Shen et al. [10] divide abstraction into structural and semantic parts, and present a visual analytics tool, OntoVis, where the relations in heterogeneous networks are reduced based on the concept of network ontology. However, all three suffer from insufficiently providing egocentric views to facilitate explorations. Besides, they consider simply links in the one step neighborhood of each node. We argue that high-order topological and relational information should be modeled to produce more meaningful abstraction from diverse aspects through combining our existing methods [17] with the proposed signature profiles.

Network Skeleton. Network skeleton refers to the hidden structural backbone of the network in a macro view. They preserve various topological properties of the graph, and thus can be regarded as a kind of abstraction from the global perspective. A. Y. Wu et al. [18] use recursive graph simplification to construct a multilevel mesh, which is a reduced graph of microclusters and preserves the characteristics of scale-free networks. D. Vincent and B. Cecile [19] perform transitive reduction on directed graph data, which is an edge-removing operation aiming at retaining the reachability between nodes. They define transitive reduction as a minimal subgraph with the same transitive closure as the original graph. By detecting the overlapping maximal cliques as supernodes, N. Du et al. [20] propose to create the backbone graph of the supernodes using the minimum spanning tree algorithm. Though the network skeleton approaches can simplify the network to some extent, it is unclear how their methods can be adopted to incorporate heterogeneous information.

Mining in Heterogeneous Networks. While most existing social network analysis studies concentrate on the homogeneous networks, some efforts are gradually shifted to the heterogeneous networks recently. D. Cai et al. [3] try to detect the community structures based on user-specified relations with importance weighting in heterogeneous social networks. They tackle this problem through learning an optimal linear combination for user-given relations to find the most relevant heterogeneous network structures. J. Zhang et al. [15] consider the importance of entities and relations to recommend objects for users in a multiple layer information network. They propose a pair-wise entity learning algorithm and integrate a modified random walk mechanism to devise the recommendation method. S. Lin et al. [6] propose an unsupervised mechanism to model the heterogeneous information surrounded each entity to identify the abnormal instances and generate reasonable explanations for them in a multi-relational social network.

3 Methodology

The formal definition for egocentric information abstraction in a heterogeneous social network is given as follows.

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

Outputs: three egocentric abstracted graphs of x, each of which belongs to the subgraph of H and corresponds to one of the three proposed abstraction views, as described in 3.3.

Definition 1. Heterogeneous Social Network). 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 end vertex, respectively. The function types$(V) \rightarrow r_1, ..., r_j, r_i \in L, j \geq 1$ maps each vertex onto its set of type labels.

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, including linear and cyclic features, are automatically selected and extracted based on the surrounding network substructure of the given ego node. 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 3.1 to 3.4.

3.1 Ego-based 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 the ego node. A LCR is defined as an ordered sequence of relations starting from the ego x. The linear combination of relations can be exploited to capture different kinds of features as behaviors for the ego x. Here we divide the features of x to two categories based on the characteristic of LCR_s: linear and cyclic features of LCR_s, as shown in Figure 4. Linear features are simply relational paths starting from the given ego x to one other vertex in the network. The linear features can be exploited to capture the interaction between x and its neighbors. Cyclic features represent paths that form a cycle. As shown in Figure 4, we consider three kinds of relational structures: self loop, triangle, and quadrangle, and each of which possesses certain physical meaning. Self loop implies a given node x has multiple and potentially diverse interactions with the other node. Triangle cycle can be regarded as a sign of highly impact triple that captures the three-way relationship among objects. Finally we exploit the quadrangle structure of LCR_s to highlight the intermediate mediators between nodes. We employ the cyclic features only up to quadrangle due to the computation complexity as well as the lack of physical meanings for those higher-order cycles. Note that it is possible that some nodes in the network do not contain some of the three cyclic features.

For example, assuming the path length k=2, the set of distinct LCR_s of node A_1 in Figure 2 is shown in Table 1. Each LCR can be regarded as a kind of behavior of A_1. Note that the inverse edge set E^{-1} is the set of all edges $(v_1, \wr^{-1}, v_2)$ such that$(v_2, \wr, v_1) \in E$. And we only regard the direction of edges

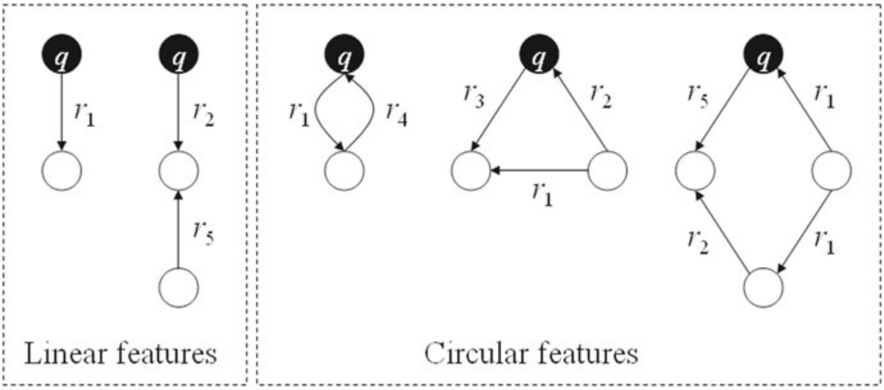


Fig. 4: The features of linear and cyclic relational structures.

Table 1: Two-steps LCRs from A1 of Figure 2.

Linear Features	LCR_1	$< hasActor^1, writeScript^1 >$
	LCR_2	$< hasActor^1, direct^1 >$
	LCR_3	$< hasActor^1, direct^1 >$
	LCR_4	$< hasActor^1, hasActor >$
	LCR_5	$< hasActor^1, originateFrom >$
	LCR_6	$< produce, direct >$
Cyclic Features	Self Loop LCR_7	$< produce, hasActor >$
	Triangle LCR_8	$< spouseOf, writeScript, hasActor >$
	Quadrangle LCR_9	$< produce, direct, direct, hasActor >$
	Quadrangle $LCR_1 0$	$< hasActor, direct, direct, hasActor >$

for linear features. For cyclic features, we do not consider the information of inverse and simply record the relations. Besides, we only record once for each cyclic LCR.

3.2 Nodes and Paths Sampling

In this section we perform certain statistic sampling on these extracted linear combination of relations (i.e., ego features) to compute the feature values. Two independent and identically-distributed (I.I.D.) random experiments are designed and applied. In the first random experiment (RE_1), we randomly select a node x from the network, then randomly select an edge e_1 starting from x, denoted by $< x, e_1, y >$, further randomly select another edge e_2 starting from y, denoted by $< y, e_2, z >$, and so forth. This stops when the number of edges chosen reaches k. The second one (RE_2) looks very similar to the first, except that we start from a randomly chosen edge $< a, e, b >$

instead of a node. Next we randomly pick another edge starting from node b. Again, this continues until k edges are chosen. The outcomes of either experiment is a path, and which we can define two random variables X and L. X represents the starting node of that path and L represents the LCR of this path. Note in this example, an instance of X is represented as x and one instance of L is $< e_1, e_2, ..., e_k >$. We use $X1$ and $X2$ to denote the starting node produced by RE_1 and RE_2, and the same for LCR_1 and LCR_2. With these four random variables, we then define two conditional probability mass functions: $P(L_1 = \wr | X_1 = x)$ and $P(X2 = x | L2 = \wr)$. We call the former *local frequency* of the ego node x, since it essentially stands for the probability that the LCR of a randomly picked path from x in face equals $\wr$. On the contrary, we call the latter relative frequency of an ego node since it represents the probability that an ego x is involved as the starting node in a given LCR $\wr$. The former is called "local" because this particular LCR is compared with other LCR_s 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 entire network.

Table 2: Conditional Probabilities of $RE_1 : P(L_1|X_1)$. ($tb\wr_{local}$)

	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 RE1 and RE2 for sufficient amount of times, it is possible to create two tables: $tb\wr_{local}$ and $tb\wr_{realtive}$ (e.g. probability values in Table 2 and 3, assuming there are only 7 LCR_s) 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.

3.3 Information Distilling

We propose two policies, frequency-based and rarity-based, to distill information from different views. Rarity and frequency basically occupy two opposite

Table 3: Conditional Probabilities of $RE_2 : P(X_2|L_2))$. $(tbl_{relative})$

	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)	0.01 (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)

ends of the spectrum, and each reveals either important or meaningful information about the ego. Frequent behaviors are generally important for pattern recognition and rare events can sometimes lead to certain novel discoveries. Combining the two views (i.e., local and relative view) and two policies (i.e., frequency-based and rarity-based), four abstraction measures can be created, as shown in Table 4. Here we abandon the relative rarity view since it does not possess an apparent real-world meaning. Below we illustrate the ideas of the rest three views via an example using the above two tables.

Table 4: The four abstraction measures from two viewpoints.

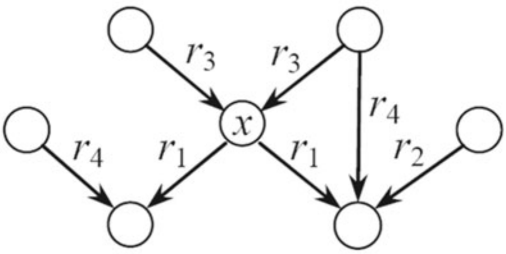
	Local	Relative
Frequency	Local Frequency	Relative Frequency
Rarity	Local Rarity	Relative Rarity

1. Local Frequency. It chooses the frequent $P(L_1|x)LCR_s$ from the vector of the given ego node x as the important ones. For example, if the threshold δ is set to $2/7$, only the top two frequent LCR_s in Table 2 $(i.e., LCR_6 and LCR_7)$ are picked to represent x_1. In other words, LCR_1 to LCR_5 are filtered out since they do not occur as frequent as other LCR_s with respect to x. The idea behind this view is that x is summarized by the frequent behaviors it involves.
2. Local Rarity. Opposite to local frequency, the rarity view of abstraction keeps the rare events happening to x and ignores the frequent ones. For the sample example δ is set to $2/7$, LCR_1 and LCR_2 will be distilled while the rest will be ruled out. Note that the "rare events" consider only those happening at least once, therefore excluding LCR_s whose conditional probabilities are zero such as LCR_3 and LCR_4. The idea behind this view is that the rare LCR_s could indicate something that should not happen but

in fact still occurs, 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. This uses Table 3 instead of Table 2. $P(X_2 = x|L_2 = \wr)$ in fact represents how frequent the ego x is involved in $\wr$ compared to other nodes. Since $\sum_X P(X_2|L_2) = 1$, we can treat each column in Table 3 as a relative comparison among all nodes for a certain LCR $\wr$. Then $P(X_2 = x|L2 = \wr)$ is representative of x if this value is relatively high compared to other nodes. In the example, LCR_3 and LCR_6 will be chosen to represent x since they are relatively high (i.e., ranked 2^{nd} and 9^{th}) compared to other nodes. The idea behind this view is that it picks the features best distinguishing x from others. Furthermore, since a heterogeneous social network generally has different types of nodes, it makes more sense to only compare the nodes of the same type when determining the rank of $P(X_2|L_2)$. For instance, it might not make sense to compare the number of publications among people from different research areas.

Note that the abstraction measures of information distilling are applied to LCRs of linear and cyclic features independently since they carry different kind of information and shall not be put on the equal ground for comparison, and usually the linear features occur more frequently than the cyclic ones since in the former case the target node is not constrained to be equivalent to the source.



(a)

ID	Ranked LCRs	Score
rs_1	$<r_1, r_4^{-1}>$	0.36 (2)
rs_2	$<r_1, r_2^{-1}>$	0.08 (5)
rs_3	$<r_3^{-1}>$	0.09 (10)
rs_4	$<r_3^{-1}, r_4>$	0.02 (79)
rs_5	$<r_1>$	0.005 (99)

(b)

Fig. 5: (a) An Example $H_{k,x}$ and (b) the ranked LCR_s.

3.4 Abstracted Graph Construction

Now we have distilled features as the abstraction for an ego node. One plausible form is to report distilled LCRs and corresponding probabilities to the users. Though it seems to be a reasonable output since $P(L_1|X_1) or P(X_2|L_2)$ can serve as a term that explains why such an abstraction is made, an alternative and more understandable way is to convert the distilled information back to a graph. Here we use an incremental method to obtain a subgraph composed of only distilled LCRs and the corresponding nodes.

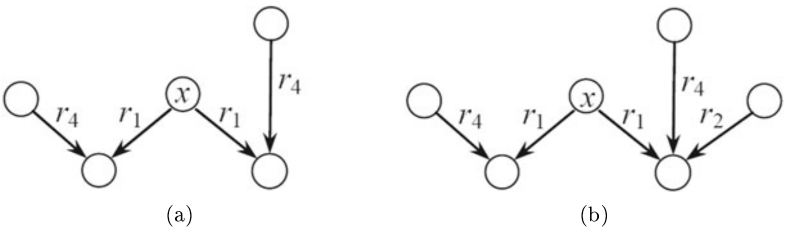


Fig. 6: The Abstracted graph after adding LCR_1 and (b) the final graph after LCR_1 and LCR_2 are added.

Figure 5 and 6 illustrate this idea. Assume we want to keep the top two scored LCRs and filter out the rest. The LCR with the highest score (i.e., LCR1) is first used to match the original network to obtain a subgraph that originates from the ego x and contains all nodes and edges involved in LCR1 (see Figure 6(a)). The same action is performed for LCR2. The final egocentric abstraction of x is shown in Figure 6(b). Note that it is not feasible to produce the abstracted graph by removing the discarded LCRs since edges involved in one LCR might also occur in others. Therefore, eliminating one of them will sometimes cause the informative LCRs to disappear. The complete algorithm of our egocentric information abstraction is given in algorithm 1. We first extract the k-step neighbor subgraph for the given ego node (line 1), and then perform sampling to derive local and relative tables (line 3-9). According to the three designated viewpoint of abstraction, the most relevant linear combinations of relations are picked (line 10-18). Finally, the abstracted graph is constructed incrementally (line 19-23).

4 Evaluations

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 the proposed features using

Algorithm 1 Egocentric Information Abstraction

Input: H: a heterogeneous network; x: the query ego node; k: the step size for linear combination of relations; δ: the information filtering threshold; *view*: policy for information distilling; *feature_ option*: option for the linear and cyclic features.
Output: H^{abs}: the abstracted graph from different views

Extract the k-step neighbor subgraph $H_{k,x}$ of x.
LCR = retrieve LCRs for *feature_ option* from x.
derive the table of local measure
$tbl_{local} = P(L_1|X_1)$ using LCR.
derive the table of relative measure and rank each column
$tbl_{relative} = P(X_2|L_2)$ using SP.
for j = 1 to $|LCR_s|$ **do**
  Compute the ranked value of $tbl_{relative}(:, \mathrm{j})$ in descending order.
end for
distilledSet = { }. // collect the LCRs of top score of specified view
if view = "localFrequency" **then**
  $distilledSet = distilledSet \cap argmaxOfTop\delta(tbl_{local}(x, LCR_i))$.
else if view = "localRarity" **then**
  // note that those scores equal to zero are ignored
  $distilledSet = distilledSet \cap argminOfTop\delta(tbl_{local}(x, LCR_i))$.
else if view = "relativeFrequency" **then**
  $distilledSet = distilledSet \cap argmaxOfTop\delta(tbl_{local}(x, LCR_i))$.
end if
Let H^{abs} = NULL.
for lcr *distilledSet* **do**
  instances = Find path instances in $H_{k,x}$, whose LCR equals to lcr.
  $H^{abs} = H^{abs} \cap$ instances.
end for
return : H^{abs}

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.

4.1 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 using linear and cyclic features respectively. The UCI KDD movie dataset [5] is used to construct the heterogeneous network. It contains 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 take "Meg Ryan", a famous actress, as the ego node to demonstrate the egocentric abstracted graphs for linear features of LCR. Also we choose "Tom Cruise", a famous actor, as another ego to present the egocentric abstracted graphs for cyclic features of LCR because this ego has more cyclic features. Besides, we would like 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 1. This is not a trivial network analysis since there are 116 nodes, 137 edges and 18 different LCRs.

Using the linear features, the abstracted graph of local frequency for Meg Ryan is shown in Figure 7, which captures the regular behavior of her. The filtering threshold $\delta = 20\%$ 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.

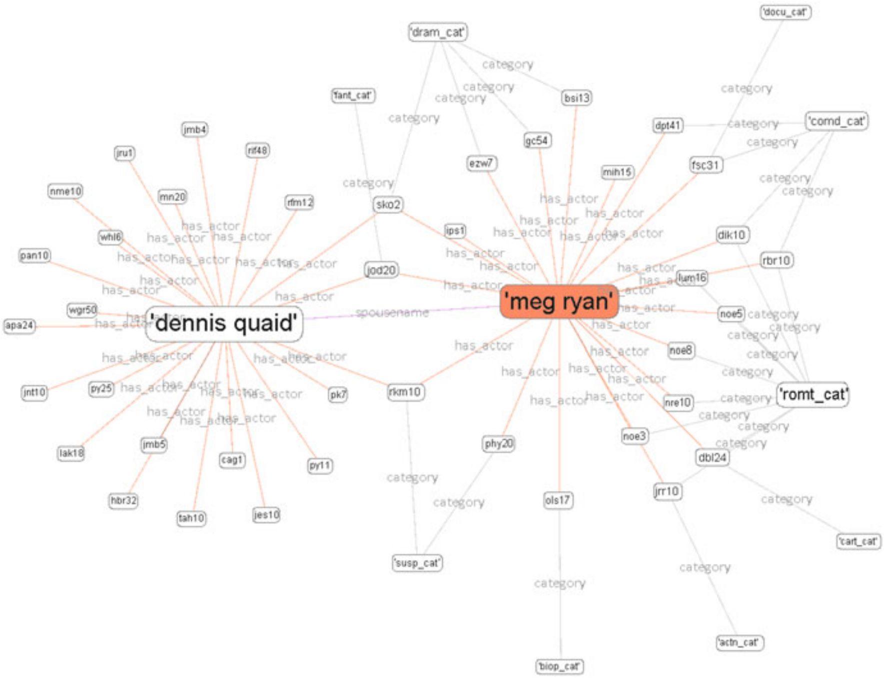


Fig. 7: Local frequency of "Meg Ryan".

Using the linear features, the local rarity view for Meg Ryan is shown in Figure 8. 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.

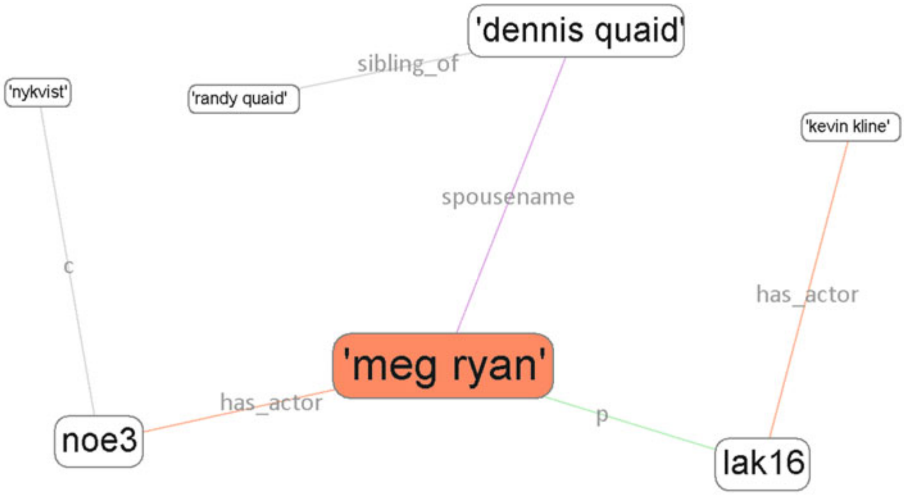


Fig. 8: Local rarity of "Meg Ryan".

Using the linear features, the abstracted graph of relative frequency for Meg Ryan is shown in Figure 9, 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 frequent 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).

Using the cyclic features, the abstracted graph of local frequency for Tom Cruise is shown in Figure 10. We can observe a set of frequent quadrangle cycles, namely $< hasActor, category, category, hasActor >, < hasActor, hasActor, hasActor, hasActor >, < hasActor, p, p, hasActor >, and < hasActor, d, d, hasActor >$. The former two shows that Cruise acted in many movies of identical categories, and collaborated with many actors in different

Fig. 9: Relative frequency of "Meg Ryan".

movies. The latter two indicates that he frequently acted in movies directed by the same person, and frequently acted in movies that has the same producer. Besides, sometimes he preferred to incorporate with permanent directors and producers (e.g., the movie "tos5" and "tos10"). Using the cyclic features, the abstracted graph of local rarity for Tom Cruise is shown in Figure 11. First, he produced and acted in one movie (i.e., "bdp30"), which is not as frequent for him. Second, there are two cyclic structures of quadrangle: (1) Tom Cruise played in the same movie with his spouse's sibling (i.e., "Paul Abbott"), (2) it is quite rare for him to involve in two movies which is produced and directed by the same person (i.e., C. Crowe).

Using the cyclic features, the abstracted graph of relative frequency for Tom Cruise is shown in Figure 12. This shows that comparing with other actors in the network; it is quite frequent for him to act in the same movie with his spouse (i.e., Nicole Kidman). Besides, the movie "bdp30" appears again to reveal that in this dataset not many actors acted in a self-produced movie.

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

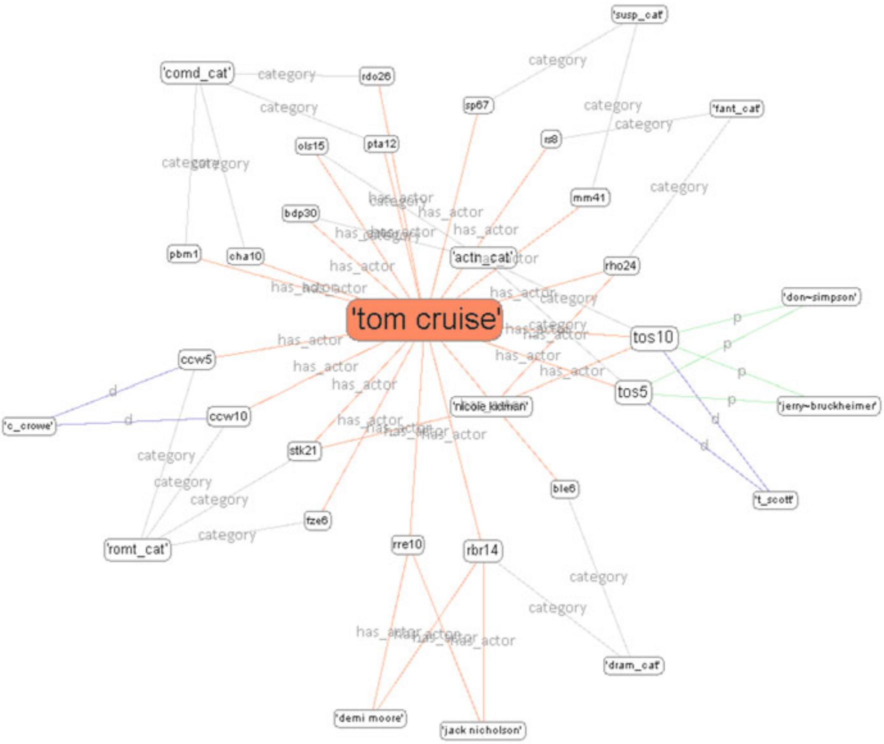


Fig. 10: Local frequency of "Tom Cruise."ʹ

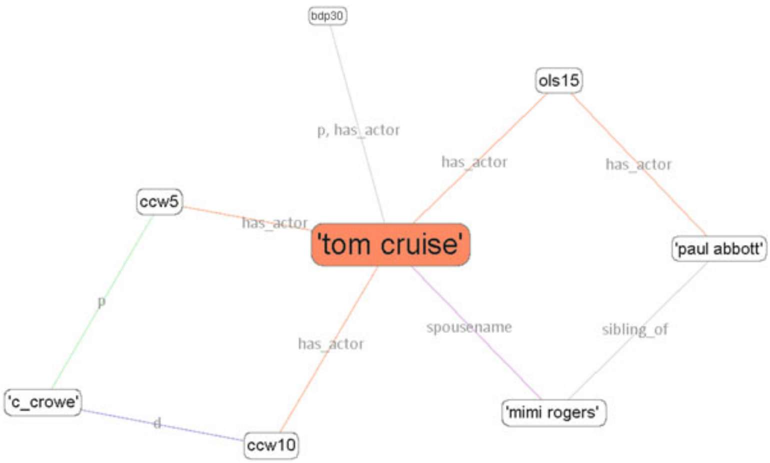


Fig. 11: Local Rarity of "Tom Cruise."ʹ

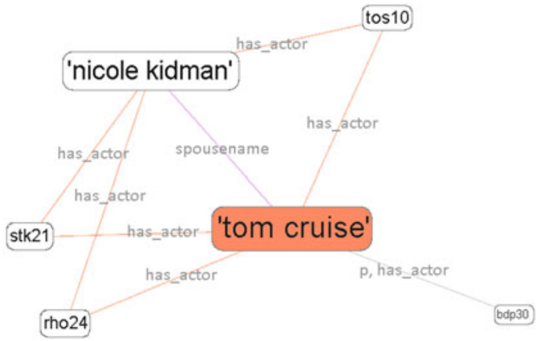


Fig. 12: Relative frequency "Tom Cruise."'

the ego node. It might even satisfy some hard-core fans by revealing certain information about her ex-husband.

4.2 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: we first choose 10 plausible gang nodes among which three were truly involved in the highest level events(i.e., gang war and industry takeover). For each gang node, three different views of egocentric abstracted graphs were generated. Together with the original k-neighborhood graph (we choose k=3 in this experiment), we will have four different set of networks (each contains 10 independent networks corresponding to 10 plausible gangs) presented to the subjects.. To avoid interference among different tasks, the IDs of all candidate gangs are randomly given for each task. These four sets of resulting graphs are shown to a total of 20 human subjects (they were not told 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 for each set. Therefore, we can examine how many candidates were picked correctly for each set. Before the experiment, the subjects were asked to study the background knowledge of this domain so they understood the meaning of each relation and the node types as well as the meaning of the events. The four generated graphs of one criminal node are illustrated in Figure 13 to 16, which are corresponding to the original 3-neighborhood graph, local frequency, local rarity, and relative frequency in order. Note that the filtering threshold δ is set to 0.2, which implies we only keep 20% of the LCRs during abstraction. The black nodes are nodes representing criminal candidates.

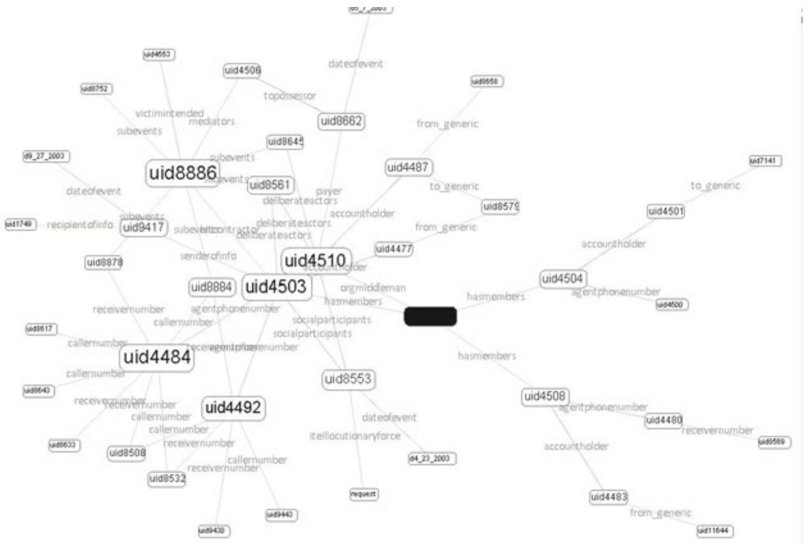


Fig. 13: The original 3-neighborhood graph.

The results are displayed in Table 5. We also show the improvement over k-step neighbor subgraph in the first column and 95% confidence interval for average time and confidence.

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

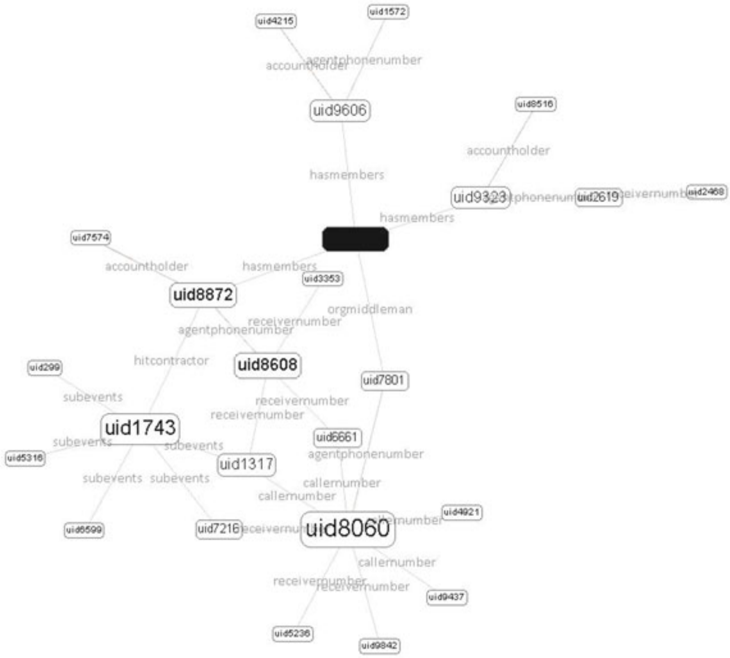


Fig. 14: Abstracted graph of local frequency.

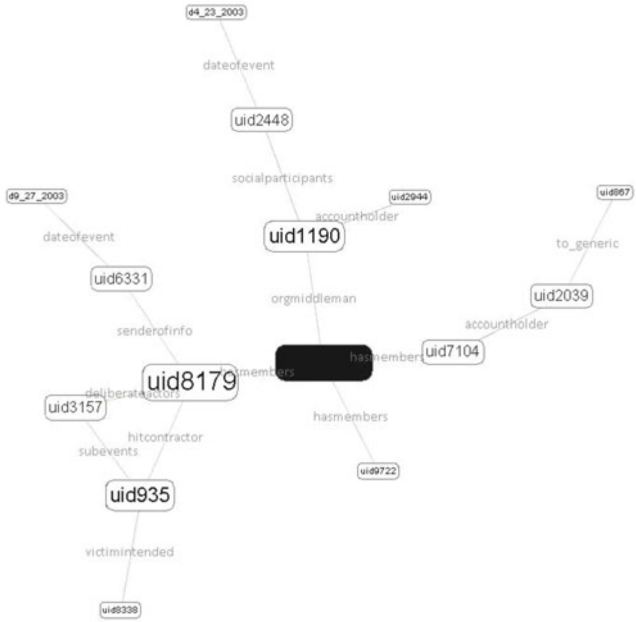


Fig. 15: Abstracted graph of local rarity.

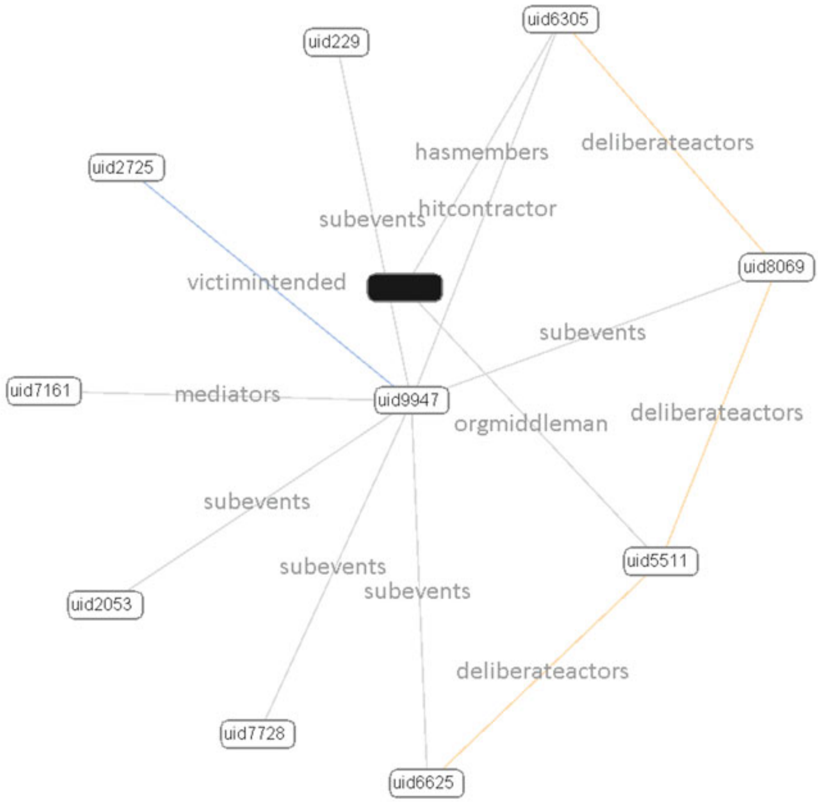


Fig. 16: Abstracted graph of relative frequency.

Table 5: Conditional Probabilities of $RE_2 : P(X_2|L_2))$. $(tbl_{relative})$

		Avg. Precision	Avg. Time (minutes)	Avg. Confidence (1∼5, 5 is the highest)
No Abstraction	k-neighborhood Graph	39/60	36.6 ± 6.6	3.15 ± 0.36
Using Abstraction	Local Frequency	41/60 (+3.3%)	18.9 ± 5.9	3.20 ± 0.35
Using Abstraction	Local Rarity	44/60 (+8.7%)	13.9 ± 3.7	3.45 ± 0.33
Using Abstraction	Relative Frequency	47/60 (+13.3%)	10.9 ± 2.2	3.73 ± 0.39

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.

5 Discussions

There are several issues worthy of further discussions:

1. 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.
2. Parameters. There are two parameters to control the level of abstraction: the k in k-neighborhood and 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 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 based on the number of different link types.
3. 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 create 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 keep important or relevant information instead of discarding 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 into a vector-space representation using symbolic modeling and sampling techniques. The reason to perform such transformation is that now we are then 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.

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.
17. C. T. Li and S. D. Lin. Egocentric Information Abstraction for Heterogeneous Social Networks. In Prof. of Intl. Conference on Advances in Social Network Analysis and Mining (ASONAM'09), 255-260, 2009.
18. A. Y. Wu, M. Garland, and J. Han. 2004. Mining Scale-free Networks Using Geodesic Clustering. In Proc. of ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining (KDD'04), 719-724.
19. D. Vincent and B. Cecile. 2005. Transitive Reduction for Social Network Analysis and Visualization. In Proc. of IEEE/WIC/ACM Intl. Conference on Web Intelligence (WI'05), 128-131.
20. N. Du, B. Wu, and B. Wang. 2007. Backbone Discovery in Social Networks. In Proc. of IEEE/WIC/ACM Intl. Conference on Web Intelligence (WI'07), 100-103.