Centrality Analysis, Role-based Clustering, and Egocentric Abstraction for Heterogeneous Social Networks

Cheng-Te Li and Shou-De Lin

Graduate Institute of Networking and Multimedia,
National Taiwan University, Taipei, Taiwan

{d98944005, sdlin}@csie.ntu.edu.tw

ABSTRACT

The social network is a powerful data structure allowing the depiction of relationship information between entities. Recent researchers have proposed many successful methods on analyzing homogeneous social networks assuming only a single type of node and relation. Nevertheless, real-world complex networks are usually heterogeneous, which presumes a network can be composed of different types of nodes and relations. In this paper, we propose an unsupervised tensor-based mechanism considering higher-order relational information to model the complex semantics of a heterogeneous social network. Based on the model we present solutions to three critical issues in heterogeneous networks. The first concerns identifying central nodes in the heterogeneous network. Second, we propose a role-based clustering method to identify nodes which play similar roles in the network. Finally, we propose an egocentric abstraction mechanism to facilitate further explorations in a complex social network. The evaluations are conducted on a real-world movie dataset and an artificial crime dataset with promising results.

Keywords: social network, heterogeneous information networks, centrality, clustering, egocentric, information abstraction.

1. INTRODUCTION

A social network is a graph in nature, where the nodes stand for actors (e.g., authors and websites) and the edges between two actors represent their relationships (e.g., co-authorship and referral). In social network analysis (SNA), people have proposed different measures for the graph structure to model some general phenomena or to capture some hidden properties, like the well-known small-world phenomena [42]. Analyzing a social network can not only assist experts in understanding the social phenomenon but also help laymen manage their social circles.

Identifying central nodes and performing entity clustering are two major research directions in social network analysis. There are already many centrality measures proposed for different types of networks. For example, the degree centrality is used to determine the importance of an author or a paper in the bibliography network [41]; the eigenvector centrality has been applied to estimate the importance of a website in the World-Wide-Web [16][28]; and the betweenness centrality has been utilized to identify crucial persons connecting multiple departments in an organization [14]. On the other hand, entity clustering aims at grouping nodes sharing some common characteristics into several clusters. In the context of social network mining, entity clustering can be divided into two categories. The first is to find *communities* or their structures [25]. A community is a subgraph containing tensely intra-connected edges within it and loosely inter-connected edges across communities. The second is to determine the *network positions* (or *social roles*) [41] of entities playing similar roles or having close semantics in the network. This paper focuses on the second type of clustering in social networks.

Although there are already various successful proposals for centrality measures and social position analysis, 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 [41]. For

example, both the Web and the citation graph (i.e., nodes are authors and edges represent co-authorships) 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, termed heterogeneous social network [41], has been proposed to describe the complex relationships (i.e., typed edges) among entities. For example, a heterogeneous movie network shown in Figure 1 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, *has actor*, A_1>, <M_3, *originate from*, 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.

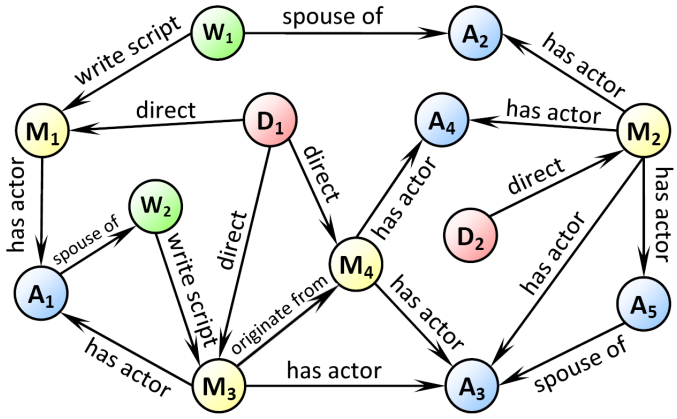


Fig. 1. A heterogeneous social network for movie domain. The capital letter of each node stands for its type: M(movie), D(director), A(actor), and W(writer). Besides, there are five relation types, including "write script", "has actor", "spouse of", "direct", and "originate from" in this example.

A heterogeneous social network in fact contains significantly more semantic information than a homogeneous one. Therefore, applying homogeneous analysis methods to it could lead to a loss of information in the process. For example, in degree centrality, a person with multiple *distinct* types of interactions with others is treated identically as the one with multiple *identical* type of interaction with others, as long as they have identical number of edges. Indeed people might believe the former is more central since it connects diversely to others. Therefore, we think it is necessary to consider the typed labels of links or nodes in heterogeneous social network analysis.

Similar issue occurs for entity clustering in a heterogeneous social network. We believe it is better to take both the typed relations and the topological information into account for clustering. Let us consider an example illustrated in Figure 2 of a heterogeneous social network where different kinds of edges (i.e., line, dot and dash) indicate different types of relations. The three gray regions are the results of entity clustering exploiting a conventional community detection algorithm [26]. We can observe it simply uses structural information to find densely inter-connected groups and ignores the heterogeneous information provided through the labels. In this paper we propose a role-based clustering method to identify nodes playing the similar semantic role in a heterogeneous social network. For the example in Figure 2, different labels on nodes indicate the diverse role-based clusters, such as the two nodes marked as Z are more like outliers involving in only a dotted relation; the three nodes marked as X are involved in three different relations. Nodes lie in different communities can still play similar roles.

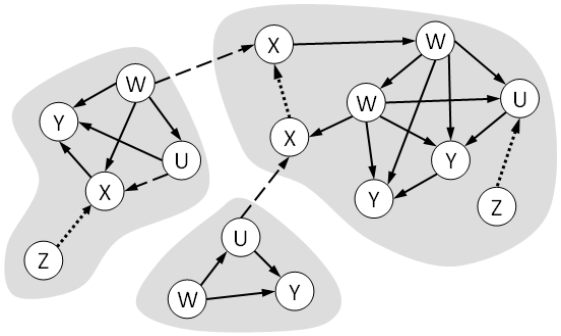


Fig. 2. An illustrated example for the community detection and the proposed role-based clustering.

On the other hand, it is known that a heterogeneous social network usually contains thousands or even millions of nodes and links with corresponding semantic labels. For example in Figure 3, it shows the overwhelming and complex information usually hinder further human observing. Thus, for the purpose of facilitating further exploration, we propose the egocentric abstraction task for heterogeneous social networks trying to identify an abstracted network structure surrounding a given node (denoted as *ego*). Such abstraction can assist users in finding solutions for questions relevant to a specific node such as "what are the normal and special behaviors of the node *x*" and "where are the differences between the node *x* others."

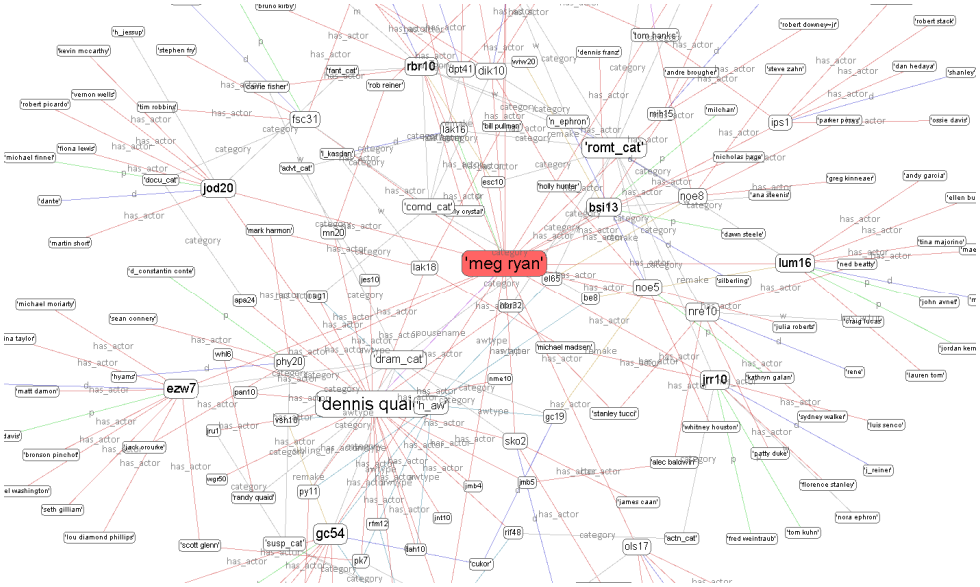


Fig. 3. An illustrated visualization of heterogeneous social network using UCI KDD movie dataset [13]. This is the two-step neighborhood graph from the famous movie actress "Meg Ryan".

We summarize the four major contributions in this paper:

1. ***A general framework for modeling the semantics of nodes.*** We suggest integrating the high-order relational information with the graph topology to model the semantics of nodes. We thus propose a tensor-based relational algebra to capture the neighborhood information of a node in an unsupervised manner. By doing this, we can then transform a heterogeneous social network into a propositional format and facilitate many mining tasks such as the following. Comparing with previous models as will be introduced in next section, our model is more general and does not require inputs from domain experts.

2. ***Determining the centrality of nodes.*** To extend the concepts of centrality from the homogeneous realm, we propose three measures, namely contribution-based, diversity-based, and similarity-based centralities to identify central nodes in a heterogeneous social network. Each measure delivers a certain semantic meaning within the complex realm. To our knowledge, this is the first attempt to extend the idea of centrality to heterogeneous social networks.
3. ***Clustering nodes based on their roles.*** Instead of grouping nodes according to the principles of community (i.e., nodes densely connected should be put together), we propose a method to cluster nodes in a heterogeneous social network based on their social positions in the network. Nodes playing similar semantic roles are grouped and the system can further provide explanation for validation.
4. ***Egocentric information abstraction.*** For any user-specified ego node, we propose an unsupervised method to summarize its ego-based subgraph from different views, including common or unique behaviors compared to itself and to others. To our knowledge, both the problem and the solution are novel. We perform experiments on both artificial and natural dataset to demonstrate the benefit of our system.

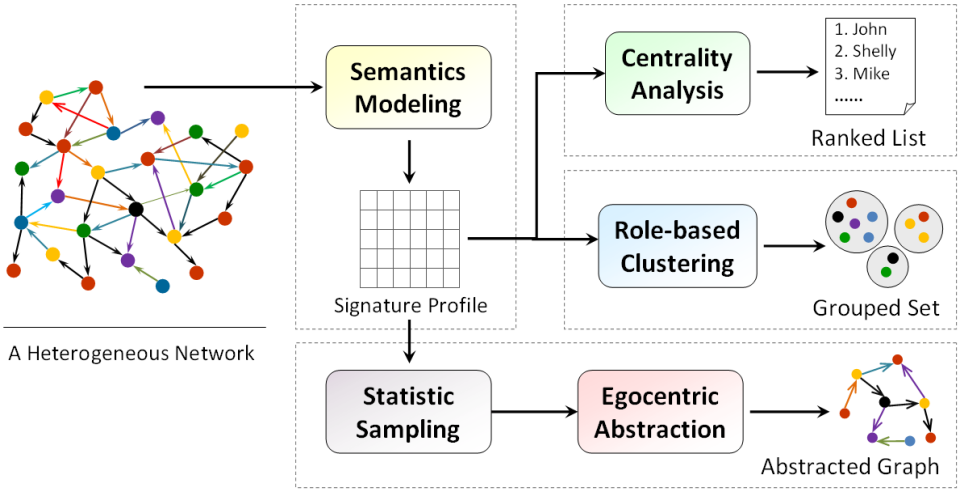


Fig. 4. The high-level framework for mining tasks in this paper.

The overall framework is illustrated in Figure 4. Given a heterogeneous network, in which different colors represent distinct types of nodes and relations. The four dotted rectangular areas correspond to the above four objectives, including modeling, centrality, clustering, and abstraction. The solutions to the second and third parts are directly based on the signature profile produced through modeling, and their outputs are a ranked list, grouped sets. For the last part, we first perform some statistic sampling using the profiles, and then provide egocentric abstraction from diverse views. The outputs are abstracted graphs from different viewpoints.

In details, our framework starts from a series of definitions about the relation sequence, relation sequence set, and relation sequence matrix as well as the operations on them. These establish the cornerstone of modeling the high-order relationship information of a heterogeneous social network. Then, we introduce the relational adjacency matrix which encodes the directly connected neighbors of each node. By applying the proposed operations, it is possible to further construct the *k*-step relational adjacency matrix to capture the indirectly-connected relational paths between any two entities. Eventually, a 3rd order relational adjacency tensor is proposed to model the topological and relational information, and based on it we can retrieve the signature profile for each node. These signature profiles essentially transform the nodes from the graph form to a matrix or vector-space representation, which enables us to exploit existing data mining techniques on it. Then we propose three heterogeneous centrality measures to identify central

nodes, namely contribution-based, diversity-based, and similarity-based centrality. We also present a role-based entity clustering method based on the signature profiles. Finally, three abstraction views, namely local frequency, local rarity and relative frequency views are proposed to serve as the distilling criteria for egocentric information abstraction.

In what follows, we will start by briefly reviewing some existing studies related to centrality measures, network clustering, and network abstraction in Section 2. We describe our model in Section 3. In Section 4, we define three heterogeneous centrality measures, and propose the role-based entity clustering. The method to perform egocentric information abstraction is developed in Section 5. Section 6 elaborates the evaluations for all proposed solutions on a movie and a crime dataset. We conclude in Section 7.

2. RELATED WORK

The study of social network mining has proceeded for more than ten years. We would like to review some works related to issues we intend to handle. Here we categorize them into the following topics: node centrality, network clustering, and graph abstraction.

2.1 Node Centrality

In sociology and graph theory, many centrality measures have been defined to measure the importance of nodes in a network based on the structural connectivity. Here, we first review some common centrality measures for homogeneous social networks.

- *Degree Centrality*. It defines the centrality of nodes as the degree of them. L.C. Freeman [9] If a node has more directly connected neighbors, it is regarded as an active individual.
- *Closeness Centrality*. Nodes tending to have shorter geodesic distance to others will have higher closeness scores [9]. The measure is defined as $C_C(x)=1/\Sigma_{y \in V} dist(x,y)$, where $dist(x,y)$ is the length of the shortest path between node x and y.
- *Betweenness Centrality*. It measures whether a node plays the bridging role in a social network. If removing a node leads to the destruction of many shortest paths between pairs of nodes, this node will be regarded as having a higher betweenness centrality score [10]. The betweenness centrality can be generated as $C_B(x)=\Sigma_{s \neq x \neq t \in V,\ s \neq t} \sigma_{st}(x)/\sigma_{st}$, where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(x)$ is the number of shortest paths from s to t that pass through node x.
- *Eigenvector Centrality*. Suggested by P. Bonacich [2], eigenvector-based centrality assigns a higher score to nodes connecting to other highly centralized nodes. The definition is $C_E(x_i)=\Sigma_{j=1\sim n} a_{ij} C_E(x_j)$, where a_{ij} is the (i,j) element in adjacency matrix. This directly implies the well-known concept of the eigenvector computation $AX=\lambda X$ and the eigenvector represents the converged centrality scores. A similar idea has been applied to estimate the importance of nodes as well. The two famous measurements for Web mining, PageRank [28] and HITS [16], are simply realizations of eigenvector centrality.
- *Information Centrality*. This employs the efficiency of information propagation as the criteria to define the influence of nodes [17]. The network efficiency is given by $E_G = (\Sigma_{i \neq j \in G} \varepsilon_{ij})(n(n-1))^{-1} = (n(n-1))^{-1}(\Sigma_{i \neq j \in G} d_{ij}^{-1})$, where $n=|V|$, the efficiency ε_{ij} is the cost of communication between node i and j, and it is equal to the inverse of the geodesic distance d_{ij}. Then the information centrality for node x can be defined as when the edges connected to x are removed, what is the relative drop of the network efficiency: $C_I(x) = (E_G - E_{G/x})/E_G$, where G/x is the network without the edges involved in x.

The above measures significantly rely on the network topology. However, they cannot be applied effectively to the heterogeneous social networks due to the ignorance of the relational information. J. Shetty et al. [34] propose the event-based graph entropy to find important nodes in the Enron email action graph by utilizing the relationship information. However, their method assumes a labeled graph with a hierarchical structure, and focuses only on finding a kind of

leader or bridging individual. Besides, their model relies on temporal information about the interactions among nodes, and thus cannot be employed in the static network directly. In contrast, our approach is more general and can be applied to any static heterogeneous networks. M. Barthelemy et al. [3] propose some statistic measures to estimate the extent of semantics for nodes and types of nodes in a multi-relational graph. However, they only consider the one-step linkages and their method needs prior knowledge about the semantic graph (i.e., the type information). D. Zhou et al. [47] perform co-ranking important entities of two types through randomly walking on their respective single-relational networks and coupling them using the mutually reinforcing relationship between them. Though their method works on the so-called heterogeneous network, no labeled semantic relationships are actually utilized. The same problem occurs on J. Zhang et al. [46]. Though they devise a random walk model for ranking relevant objects in a Web network with multiple types of entities, the labeled relationships information between entities are still neglected. To achieving the tasks of ranking and searching for nodes in a relational graph, E. Minkov et al. [23] consider the typed information and take advantage of a supervised learning to improve the performance of random walk model. Though the high-order relationship is used for the ranking, the supervised approach needs prior knowledge about the network configuration, and thus not easily to apply to any heterogeneous network data.

2.2 Network Clustering

The term clustering means grouping nodes sharing common characteristics. The pre-specified definition of such characteristics usually determines the results. There are two major directions for node clustering in a network: namely to identify *community structure*, and *social positions* of nodes. For the former, the basic idea is to group nodes based on a graph topology principle, which states "clustered nodes are those tensely intra-connected in the graph structure while some loosely inter-connected nodes locate between clusters" [26]. On the other hand, social position-based clustering groups the nodes based on the local structural patterns. That is, if two nodes have similar neighbors or have similar connections to others, they should be put together, regardless of whether they are from the same community. What follows is a brief introduction for existing methods on these two topics.

There are several works related to detecting communities in a homogeneous social network. The general approach to find dense subgraphs is by partitioning the graph recursively M. Field [8]. KL algorithm [15] maximizes a benefit function to partition the graph greedily. Recently, researchers have proposed the modularity-based approach [6][12][26][27] for detecting communities. The idea behind modularity is to ensure the number of edges across groups is not only small but also smaller than expected. Besides, W. Hwang et al. [14] propose the bridging centrality integrating the global and local features to identify bridges between communities, and then removes some edges from the network by the Bridge-Cut algorithm to form several cohesive subgraphs. SCAN algorithm [45] defines structural similarity as the base to present a density-based structural clustering in a bottom-up manner. V. Satuluri et al. [30] propose utilizing stochastic flows for community detection. Despite their great success in homogeneous networks, none can be easily adopted in the heterogeneous domain.

For heterogeneous social networks, the definition of community can differ from homogeneous social networks. A heterogeneous community does not have to process dense connections within a certain relational graph, but its members might share similar and frequent interactions with communities of other relational graphs. Spectral relational clustering [21][22][38] is one of the most well-known approaches to identify communities in a heterogeneous network, which formulates the problem into factorization on multiple matrices or tensor structures. Then, by optimizing a certain objective function with some relational constraints and the tolerant approximation, the relational clusters can be attained. Instead of partitioning the graph regarding patterns of interaction, we aim at grouping nodes based on their roles. D. Cai et al. [5] address another kind of community detection problem in heterogeneous networks through learning an optimal linear combination of a user-queried relational structure. More recently, Y. Sun et al. [37] propose a NetClus algorithm to discover a new kind of clusters in a heterogeneous network,

where each member in the cluster is a subgraph with star-shaped schema. However, their solution is restricted to this specific schema and therefore cannot deal with higher-order relational information.

On the other hand, the social position analysis considers the neighbor structures of nodes to estimate the roles they play, which has similar goals as our social role-based clustering algorithm. Diverse social positions can be identified using various *equivalence classes*. There are three major kinds of equivalence class, including structural, automorphic, and regular equivalence [41]. Two nodes are considered as structurally equivalent if both have identical links to and from other identical actors. The automorphic equivalence defines two actors as equivalent if they have the same local structure (i.e., pattern of graph isomorphism). The third and the least strict is the regular equivalence, which defines two actors as equivalent if both have similar links to members of other regular equivalence groups.

The existing solutions to clustering nodes based on social roles, such as the blockmodeling approach [43] and profile similarity approach R. Breiger et al. [4], suffer from the common limitation of the lack of ability to handle multiple relational data, let alone considering higher order connections. That is, past methods simply exploit the statistics of the immediate nodes and links for equivalence detection. Besides, J. Scripps et al. [32] identify different roles of nodes in a community, including ambassadors, big fish, loners, and bridges for homogeneous social networks. To the best of our knowledge, our solution to group nodes based on their semantic roles is the first to address this problem in the heterogeneous realm. It can be regarded as a generalized social role discovery model which simultaneously takes the diverse labeled relations and their higher-order combinations into consideration. D. Rogers et al. [29] propose the extended connectivity fingerprints as a kind of *k*-step traversals on molecular graph for substructure and similarity search. Lin et al. [19][20] propose a model to capture the semantics of nodes in a multi-relational network to identify nodes with abnormal behaviors. Our tensor based model can be regarded as a theoretically more sound, intuitive, and general model which can naturally adopt higher-order interactions as well as temporal information (by adding one more dimension in the tensor).

2.3 Graph Abstraction and Summarization

We further divide this research theme into three sub-categories:

- *Graph Summarization*. It is about generating the compact summarized representation for a large graph. L. Zou et al. [48] 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. [39] 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. [24] 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. [35] 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. Shen et al. [33] 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 the existing abstraction ideas [18][49] with the proposed signature profile model.

- *Network Skeleton*. This refers to the hidden structural backbone of the network in a macro view. Network skeletons preserve various topological properties of the graph, and thus can be regarded as a kind of abstraction. A.Y. Wu et al. [44] 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 et al. [40] perform transitive reduction, which is an edge-removing operation without losing reachability between any two nodes, on directed graph data. 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. [7] build the backbone graph of the supernodes using the minimum spanning tree algorithm, where the amount of overlap serves as the distance between them. Though the above results simplify the network to some extent, it is unclear how their methods can be adopted to incorporate heterogeneous information.

3. MODELING HETEROGENEOUS NETWORKS

This section discusses our model for heterogeneous social networks. Our fundamental assumption is the information about a node has already been encoded in the form of a heterogeneous social network, and the semantics can be captured and formulated using the surrounding relational structures. We aim at profiling each node to a vector-based representation in an unsupervised manner, which automatically extracts relational features and measures the relatedness as feature values between the node and the corresponding features. The storyboard of the proposed semantics modeling for heterogeneous networks is shown in Figure 5. We first propose relation sequences as the features and define some data structures and operations on them. Using these definitions, all the relational information in the network will be described in a relational adjacency tensor. Finally, the vector-based signature profile for each node can be derived from the tensor.

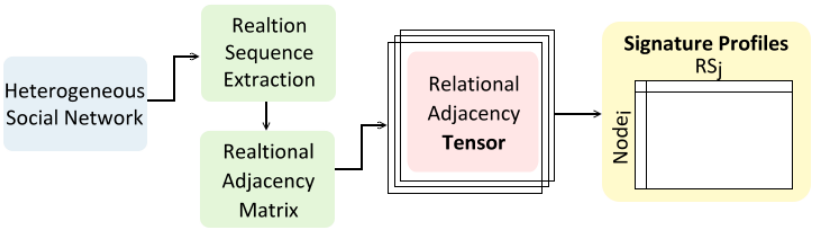


Fig. 5. The storyboard of semantics modeling.

3.1 Problem Definition

A heterogeneous network is composed of a topological part and relational part. Each node can be characterized using its neighborhood which consists of a set of directly or indirectly connected nodes and links.

Definition 3.1 (*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) $\rightarrow \{\{r_1,\ldots,r_j\}, r_i \in L, j \geq 1\}$ maps each vertex onto its set of type labels.

The goal of our modeling is to transform a multi-relational graph to a vector-based representation without prior user knowledge, namely using an unsupervised method.

3.2 Relational Adjacency Matrix

As already mentioned, the role of each node is encoded by its relational neighborhood. This motivates our idea of defining the *k*-step relational adjacency matrix to capture the direct and indirect relationships between nodes. We start from some basic definitions of relational data structures.

Definition 3.2 (*Relation Sequence*). A sequence of labeled relations is called a *relation sequence* (RS). A *k*-step relation sequence (k>0) is defined as a sequence of k labeled relations $<r_{x1},r_{x2},\ldots,r_{xk}>$ where each $r_{xk} \in L$.

Definition 3.3 (*Relation Sequence Group*). The group of relation sequences $\{RS_1,RS_2,\ldots\}$ is called a *relation sequence group* (RSG). Note that RS_i can be of any length, and duplicate elements can exist in an RSG. We can represent the duplicate elements in an RSG using a numerical number before each occurrence of distinct RS. For example, $\{3RS_1,1RS_2,\ldots\}$ means in this group there are three RS_1 and one RS_2. The counting is important since later we will show how it can be treated as the feature values in the signature profile.

Definition 3.4 (*Relation Sequence Matrix*). A relation sequence matrix RSM is defined as an $n \times n$ matrix and each element of the matrix is a relation sequence group. In our model the constant n stands for the number of nodes in the social network.

Then we define the multiplication and summation operations between two relation sequences and two relation sequence groups.

Definition 3.5 (*Multiplication on Two Relation Sequences*). Given two relation sequences $<r_{x1},r_{x2},\ldots,r_{xi}>$ and $<r_{y1},r_{y2},\ldots,r_{yj}>$, their multiplication (denoted by "×") is defined as concatenating the second sequence after the first one as $<r_{x1},r_{x2},\ldots,r_{xi},r_{y1},r_{y2},\ldots,r_{yj}>$. Note that this operation is not symmetric.

Definition 3.6 (*Multiplication on Two Relation Sequence Groups*). Given two relation sequence groups $RSG_a=\{RS_{a1},RS_{a2},\ldots,RS_{an}\}$ and $RSG_b=\{RS_{b1},RS_{b2},\ldots,RS_{bm}\}$, their multiplication is defined as the group of all pair-wise relation sequences multiplied from both groups, as the following equation describes. If either RSG is an empty group, then the resulting relation sequence group is also empty.

$$RSG_a \times RSG_b = \{ \underset{i=1\ldots n}{\forall} \underset{j=1\ldots m}{\forall} RS_{ai} \cdot RS_{bj} \} \tag{1}$$

Definition 3.7 (*Summation of Relation Sequence Groups*). Given multiple relation sequence groups $RSG_1=\{RS_{11},RS_{12},\ldots,RS_{1p}\}$, $RSG_2=\{RS_{21},RS_{22},\ldots,RS_{2q}\}$,..., $RSG_m=\{RS_{m1},RS_{m2},\ldots,RS_{mr}\}$, their summation (denoted by $RSG_1+RSG_2+...+RSG_m$) is defined as the group of all elements in every RSGs. That says, $RSG_1+RSG_2+...+RSG_m=\{RS_{11},\ldots,RS_{1p},RS_{21},\ldots,RS_{2q},RS_{m1},\ldots,RS_{kr}\}$.

Since each element in a RSM is a RSG, and we have defined the multiplication and summation for RSG, the multiplication of two RSMs can be defined as similar to the multiplication of two numerical matrices.

Definition 3.8 (*Multiplication of Two Relational Sequence Matrices*). Given two relation sequence matrices RSM_a and RSM_b, and assuming e^a_{ij} and e^b_{ij} represents the element (which is an RSG) of the i^{th} row and j^{th} column in each matrix respectively. Let $RSM_{ab}=RSM_a \times RSM_b$, then e^{ab}_{ij}, the element of the i^{th} row and the j^{th} column of RSM_{ab}, can be derived as

$$e^{ab}_{ij} = \sum_{x=1}^{n} e^a_{ix} \times e^b_{xj} \tag{2}$$

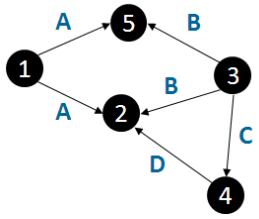
Now we can introduce the one-step and *k*-step relational adjacency matrix to model the neighbor relational structure of each node in the network.

Definition 3.9 (*One-step Relational Adjacency Matrix*). The one-step relational adjacency matrix of a given heterogeneous social network *H* (denoted by ReAM^1) is a relation sequence matrix that captures the direct adjacency relationship between any two nodes. That is, given a social network with n nodes, each element in its one-step adjacency matrix is the group of direct labeled relations connecting two corresponding nodes. Note that there can be multiple direct connections between two nodes in the network, and a node can also connect to itself given there is a self-loop.

For example, the ReAM^1 of Figure 6(a) is shown in Figure 6(b). Note that r^{-1} here represents the inverse edge of typed label *r*, as the label for the edge is (v_1, r^{-1}, v_2) such that $(v_2, r, v_1) \in E$.

Definition 3.10 (*k-step Relational Adjacency Matrix*). A *k*-step relational adjacency matrix of *H*, denoted by ReAM^k, is defined as $\text{ReAM}^{k-1} \times \text{ReAM}^1$, and can be generated incrementally from the one-step relational adjacency matrix.

For example, the ReAM^2 is derived through $\text{ReAM}^1 \times \text{ReAM}^1$. Figure 6(c) shows the 2-step relational adjacency matrix of 6(a) when multiplying 6(b) by itself.



(a)

	{<A>}			{<A>}
{<A^{-1}>}		{<B^{-1}>}	{<D^{-1}>}	
	{<B>}		{<C>}	{<B>}
	<D>	<C^{-1}>		
{<A^{-1}>}		<B^{-1}>		

(b)

		{2<A,B^{-1}>}	{<A,D^{-1}>}	
		{<D^{-1},C^{-1}>}	{<B^{-1},C>}	{<A^{-1},A>, <B^{-1},B>}
{2<B,A^{-1}>}	{<C,D>}		{<B,D^{-1}>}	
{<D,A^{-1}>}	{<C^{-1},B>}	{<D,B^{-1}>}		{<C^{-1},B>}
	{<B^{-1},B>, <A^{-1},A>}		{<B^{-1},C>}	

(c)

Fig. 6. The example of deriving (b) ReAM^1 and (c) ReAM^2 from (a) a simple heterogeneous social network.

The *k*-step relational adjacency matrix essentially captures all the *k*-step relational paths from one node to its *k*-step neighbors in the entire heterogeneous social network. Each element in the matrix represents the relational paths connecting two nodes. Each row in the matrix can be regarded as all the *k*-step outgoing paths from one node, and each column stands for all the *k*-step incoming paths into a node.

3.3 Relational Adjacency Tensor

Here we propose the *relational adjacency tensor* that integrates the relational adjacency matrices introduced in 3.2 to represent the surrounding environment of each node in a heterogeneous social network. A tensor is a generalized form of a matrix, a vector, and a scalar in the field of multilinear algebra. It is powerful for expressing high-order data. For example, the dynamic social network can be described by the time-evolved adjacency matrices [36]. In this case, a 3rd order tensor, with time being the 3rd mode, is used. DataCube such as the customer-product-branch sales data can be represented using 3rd order tensors as well.

Our modeling extends the original idea of the tensor from the numerical domain to the relational domain in social networks. The order of the proposed relational adjacency tensor is three: the first represents the source nodes, the second represents the target nodes and the third is the step size k.

Definition 3.11 (*Relational Adjacency Tensor*). A 3rd order tensor, which consists of k slices of a series of relational adjacency matrices ReAM^1, ReAM^2,…, ReAM^k, is called relational adjacency tensor, denoted by RAT^k.

The RAT is constructed by combining the ReAMs (from ReAM^1 to ReAM^k in order) as the slices along the 3rd mode of k-step relations, as illustrated in Figure 7(a) and 7(b).

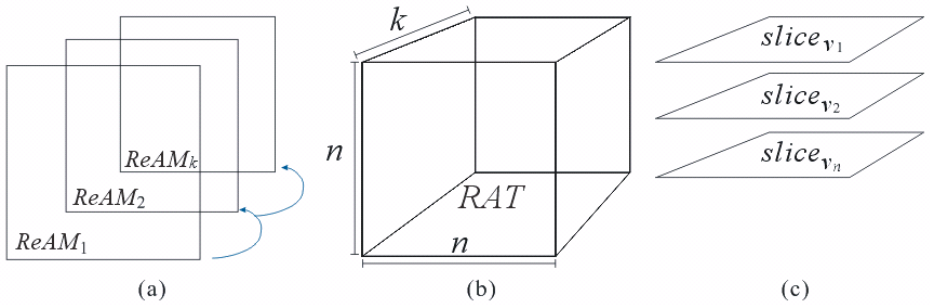


Fig. 7. (a) ReAM^k can be created progressively from ReAM^1. (b) The RAT^k is composed of the slices from ReAM^1 to ReAM^k in order (c) Each horizontal slice of RAT^k is used to construct the *signature profile* of the corresponding node.

3.4 Signature Profiles

With the RAT, we can now introduce the signature profile as the model to capture the semantics of the nodes in a heterogeneous social network. Each horizontal slice (we call a signature slice) of RAT is used to represent a single node, as illustrated in Figure 7(c). A *signature profile* (denoted as *sp*) of each node can be summarized from the corresponding signature slice. The signature profile of each node v is a vector where each element in the vector stands for a relation sequence (or signature) and its value represents the frequency of that sequence in the slice. Figure 8 displays the deriving of signature profiles of v_1 and v_3 in Figure 6(a). First, the signature slices from RAT for v_1 and v_3 are retrieved, as shown in Figure 8(a) and 8(b). Then the relation sequences inside each slice can be listed and aggregated as the signatures, as shown in Figure 8(c). Finally, the signature profile for each node can be attained in the vector-based representation from signatures and the corresponding count, as shown in Figure 8(d). Note that since the number of relation sequence in a heterogeneous social network is bounded, the vector size of each entity is also bounded.

What we essentially do up to this point is to translate the original heterogeneous network into a vector representation of nodes. The major advantage of this vector-space modeling is now we are allowed to apply many existing data mining algorithms for heterogeneous social network analysis. Also this model is succinct and modularized while the operations are trivial to implement. In fact, for certain tools such as MATLAB which possess strong capability of matrix computation, one can even directly exploit the signature slice (in fact a relational matrix) for further computation without resorting to the vector profile (in which the target information is lost during aggregation). In the next section we will show how one can identify central nodes and perform role-based clustering based on this model.

k=1:		{<A>}			{<A>}
k=2:			{2<A,B^{-1}>}	{<A,D^{-1}>}	

(a) The slice for v_1 of Figure 5.

k=1:		{<B>}		{<C>}	{<B>}
k=2:	{2<B,A^{-1}>}		{<C,D>}	{<B,D^{-1}>}	

(b) The slice for v_3 of Figure 5.

$signature_{v1}$	2<A>, 2<A,B^{-1}>, <A,D^{-1}>
$signature_{v3}$	2<B>, <C>, 2<B,A^{-1}>, <C,D>, <B,D^{-1}>

(c)

	<A>	<B>	<C>	<A,B^{-1}>	<A,D^{-1}>	<B,A^{-1}>	<C,D>	<B,D^{-1}>
v_1	2	0	0	2	1	0	0	0
v_3	0	2	1	0	0	2	1	1

(d)

Fig. 8. This example is based on Figure 6 and k is set to 2. (a) The horizontal slice for v_1. (b) The horizontal slice for v_3. (c) The signatures of v_1 and v_3. (d) The signature profile of v_1 and v_3.

Algorithm 1. Tensor-based Semantics Modeling

Input: H=<V,E,L>: a heterogeneous network; k: the step size for relation sequences.
Output: $SP(x)$: the signature profile for each node x, which is a feature vector

1: Derive the one-step relational adjacency matrix $ReAM^1$.
2: $RAT = [ReAM^1]$.
3: **for** $step$ = 2 to k **do**
4: $ReAM^k = ReAM^{k-1} \times ReAM^1$. // iteratively get k-step relational adjacency matrices
5: $RAT = [RAT, ReAM^k]$. // incrementally construct the relational adjacency tensor
6: **end for**
7: $SG = \{RAT(1{:}n, 1{:}n, 1{:}k)\}$. // collect all kinds of signatures (i.e., relation sequences)
8: SP = new **int**[n][|SG|]. // initialize the signature profiles
9: **for** $x \in V$ **do** // each horizontal slice
10: $SP(x)$ = count and the times of each signature of x from $RAT(x, 1{:}n, 1{:}k)$ and store the counts into the corresponding cell of SP.
11: **end for**
12: **return:** SP

The complete procedures for our tensor-based semantics modeling can be elaborated by algorithm 1. Given the heterogeneous network and the step size, the algorithm first progressively attains the relational adjacency tensor (line 1-6). Then all signatures are collected (line 7) to construct the signature profiles (line 8). By visiting each horizontal slice of the tensor and counting the occurrence times of each node's signature (line 9-11), the profile is produced.

4. CENTRALITY AND ROLE-BASED CLUSTERING

Two of the fundamental issues in social network analysis are to determine the centrality of nodes and to group nodes according to their characteristics. In this chapter, we intend to extend the concepts of centrality and clustering to heterogeneous networks based on the derived signature profiles.

4.1 Heterogeneous Centralities

Given the signature profile of each node, we propose three diverse centrality measures for heterogeneous social networks including diversity-based, contribution-based, and similarity-based viewpoints, each of which possesses its own physical meaning.

4.1.1 *Contribution-based Centrality*. Based on the idea an individual can be regarded as central if it significantly involves various types of social events, we propose contribution-based centrality utilizing the signature profile to find the central nodes that satisfy such a condition.

The idea of contribution-based weighting focuses on measuring the extent to which a node contributes to a specific signature, compared with other nodes in the network. The term *contribution* in this sense is a relative and global concept. The equations below formulize how the contribution-based centrality of a node x, $C_{cont}(x)$, can be computed using the signature profile,

$$contribution(x, signature_i) = \frac{freq(x, signature_i)}{\sum_{y=1}^{n} freq(y, signature_i)} \tag{3}$$

$$C_{cont}(x) = \sum_{i=1}^{|signature|} contribution(x, signature_i) \tag{4}$$

where $freq(x, signature_i)$ is the frequency of a $signature_i$ occuring in x's signature profile, and $|signature|$ is the number of possible signatures. High contribution value indicates this node plays a significant role with respect to the event represented by this signature while a low contribution indicates it is not too involved in the corresponding behavior.

In other words, we sum up the probabilities each signature occurs in x's signature profile as its centrality score. The intuition behind this idea is that each signature can be treated as a kind of interaction of an individual with others. And since the contribution implies how significantly a node participates in a particular behavior, an individual significantly contributing to various signatures can be regarded as the central one. The contribution-based centrality to some extent can be regarded as the heterogeneous version of the degree centrality of homogeneous social networks. The main difference is that the former only counts the number of one-step direct "links" while the latter considers not only the higher order information (i.e., the neighborhood of a node) but also the semantics of the relations.

4.1.2 *Diversity-based Centrality*. An individual can be regarded as a center piece in a society if it has connections to different kinds of groups or is involved in a diverse range of events. Similar to the idea of the "the strength of the weak tie" [11] which argues that the few interconnected links between different clusters could be the key to the compactness of a society, in diversity-based centrality view we believe the nodes involved in more kinds of events are central.

Here, we argue that the central nodes in a heterogeneous social network should be those involved in many different kinds of signatures, no matter how significantly they are involved. The intuition behind this idea is an individual involved in diverse kinds of events has more chance of being the center of the society and connecting different kinds of others. To compute diversity-based centrality C_{div}, we first convert the signature profile of nodes into a binary one (i.e., the values become 1 if it is not zero), and then sum up each row as the centrality score of each entity.

The diversity-based centrality is positively correlated with the traditional degree centrality. However, the major difference lies in that the latter considers the number of interactions an individual is involved in while the former only consider the number of distinct interactions. Analogize to the cases in a homogeneous social network: the degree centrality corresponds to finding nodes having many contacts while the proposed diversity-based centrality corresponds to finding nodes involved in more weak links.

4.1.3 *Similarity-based Centrality*. Motivated by the idea a focal person can usually attract individuals of similar aims and naturally becomes the center of the group, we argue that a node surrounded by many similar nodes can be regarded as a central one. We define the similarity between two nodes by the cosine similarity of the corresponding vectors of the signature profiles, which is given in the following equation (5).

$$sim(x, y) = \frac{SP(x) \cdot SP(y)}{\|SP(x)\| \|SP(y)\|} \quad (5)$$

where $SP(x)$ is the vector of signature of x. Below in algorithm 2 we elaborate the procedure to generate the similarity-based centrality score of a node x, denoted by $C_{sim}(x)$. The neighbored nodes with identical types as x are first identified (line 1-2). Then we sum up the similarity scores between x and all of its neighbors, in which the similarity is divided by the geodesic distance to x. We consider the geodesic distance as the divisor because the closer nodes are supposed to have more impact on the centrality score. This method guarantees a high centrality score of x if and only if it is surrounded by nodes whose signature profiles are similar to x.

Algorithm 2. Similarity-based Centrality

Input: SP: the signature profiles of nodes; x: the indicated node; $type(x)$: the type for node x; $k_{neighbor}$: parameter to control the size of the neighborhood of x.
Output: $C_{sim}(x)$: the similarity-based centrality score for the node x.

1: $neighborhood(x)$ = all nodes within $k_{neighbor}$ steps from x.
2: $N_t(x) = \{x \mid i \in neighborhood(x)$ & $type(i) = type(x)\}$
3: $C_{sim}(x) = \sum_{i \in N_t(x)} \frac{sim(i, x)}{r^2}$ // r is the shortest distance between i and x
4: **return:** $C_{sim}(x)$

4.2 Role-based Clustering

Using the signature profiles, nodes in a heterogeneous social network can be clustered based on the roles they play in the network. That is, we think the clustering could consider what kinds of interactions a node has with others and group nodes having similar interaction patterns with others would be grouped together, regardless of how far they are away from one another in the social network.

Unlike the community detection methods which utilize structural information such as edge betweenness, geodesic distance, modularity, and etc., we propose creating a distance matrix among nodes using the signature profiles which uniquely consider the higher-order relational information of edges. That is, the distance matrix is not determined by the distance between nodes in the social network, but the distance between their signature profiles. Then we can apply various distance-based clustering algorithms (e.g., K-Means, hierarchical clustering) to the distance matrix for different purposes. Note that another advantage of our role-based clustering method over the community-based clustering algorithm is that the node types can also be taken into account since it is possible to consider only the distance matrix containing the same type of nodes. Therefore, nodes of different types (e.g., movie and person) will not be grouped together.

5. EGOCENTRIC INFORMATION ABSTRACTION

Information abstraction generally refers to the summarization and re-organization of overwhelming, raw information into a humanly-understandable representation while still retaining the important and meaningful information. The concept of information abstraction has not yet been formally defined in heterogeneous social networks, though the essences of several works, like centralities, PageRank, clustering coefficient and degree distribution [25], are related to abstraction in some sense. However, they all suffer a major weakness of summarizing a complex network into only a few numbers with the loss of a decent amount of information about connections.

In this research, we exploit the idea of information abstraction in heterogeneous social networks. Further, given the fact that a real-world social network can contain 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 [41], the node of interests can be referred to 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.

5.1 Problem Definition

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 given node x, represents the ego, and (c) the information filtering threshold δ ($0 \leq \delta \leq 1$) to control the level of abstraction.

Find: 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.

The egocentric information abstraction has four stages. First, a set of relation sequences are extracted from the surrounding substructure of the ego node. Second, the statistic dependency measures between the features and the ego node are computed. Third, some distilling criteria are applied to remove trivial information. Finally, an egocentric abstracted graph is constructed incrementally. The elaboration of these four stages is provided in 5.2 to 5.5, the flowchart is shown in Figure 9.

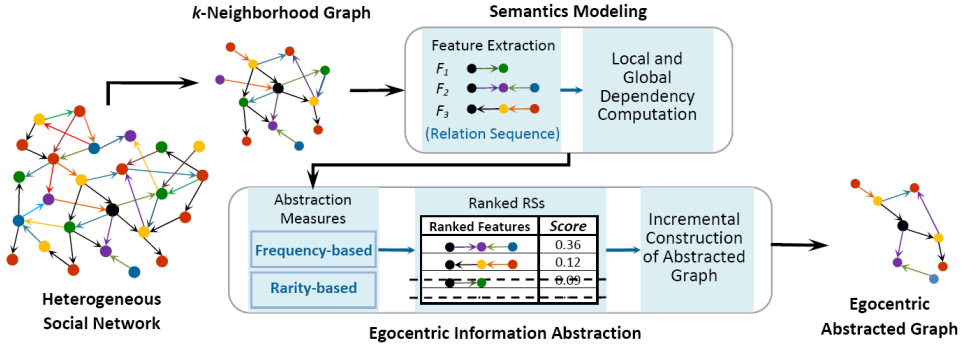


Fig. 9. The flowchart of egocentric information abstraction.

5.2 Ego Feature Extraction

We first extract the k-step neighbor subgraph $H_{k,x}$ of the ego node x. Constraining the size of the neighborhood is reasonable since it is usually assumed farer away nodes do not have as significant inferences as the closer ones. Then we propose to exploit the *relation sequence* as the base (i.e., as the ego features) to represent the surrounding structure of an ego node. For example, by taking k=2, the set of distinct relation sequences of node A_1 in Figure 1 is shown in Table I and the corresponding k-step neighbor subgraph is illustrated in Figure 10. Each relation sequence can be regarded as a kind of behavior of A_1.

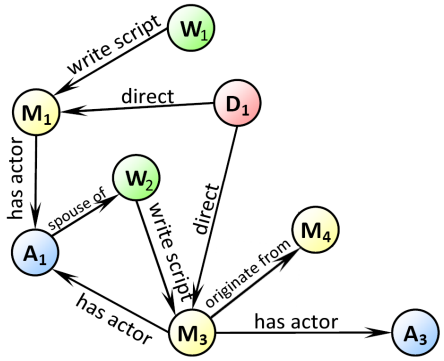


Fig. 10. *k*-step neighbor subgraph $H_{2,A1}$ for Figure 1 (k=2).

Table I. Two-steps relation sequences from A_1 for Figure 1.

rs_1	$<hasActor^{-1}, writeScript^{-1}>$
rs_2	$<hasActor^{-1}, direct^{-1}>$
rs_3	*<spouseOf, writeScript>*
rs_4	$<hasActor^{-1}, hasActor>$
rs_5	$<hasActor^{-1}, originateFrom>$

5.3 Nodes and Paths Sampling

In this section we perform certain statistic sampling on these extracted relation sequences (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 random experiment (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 based on which we can define two random variables X and RS. X represents the starting node of that path and RS represents the relation sequence of this path. Note in this example, an instance of X is represented as x and one instance of RS is $<e_1,e_2,...,e_k>$. We use X_1 and X_2 to denote the starting node produced by RE_1 and RE_2, and the same for RS_1 and RS_2.

With the four random variables, we then define two conditional probability mass functions $P(RS_1=rs|X_1=x)$ and $P(X_2=x|RS_2=rs)$. We call the former *local frequency* of the ego node, since it essentially stands for the probability that a randomly picked relation sequence from x in fact equals rs. On the contrary, we call the latter *relative frequency* of the ego node, since it represents the probability that an ego x is involved as the starting node in a relation sequence rs. The former is called "local" because this particular feature of relation sequence is compared with the other features starting from the same ego node (regardless of how it distributed in the rest of the network). The latter is called "relative" since it depends on how this feature is distributed in the entire network.

After sampling both RE_1 and RE_2 for sufficient amounts of time, it is possible to create two tables: tbl_{local} and $tbl_{relative}$ (e.g., Table II and Table III, assuming only 7 relation sequences) consisting of the corresponding conditional probabilities. We call such tables the vector-based summarization of nodes. That is, each row vector in both tables is the summarization of each node in the network. Note that in Table III the ranks of each $P(X_2|RS_2)$ compared with all nodes of the same type are listed inside the parentheses. For example, in Table III, $P(X_2=x_1|rs_4)=0$ is ranked as 99, which implies there are 99 nodes of the same type in the entire network since it possesses the smallest probability. Besides, the probability of each row sums to 1 in Table II while in Table III the probability of each column sums to 1.

Table II. Conditional probabilities of RE_1: $P(RS_1|X_1)$. (tbl_{local})

	rs_1	rs_2	rs_3	rs_4	rs_5	rs_6	rs_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

Table III. Conditional probabilities of RE_2: $P(X_2|RS_2)$. ($tbl_{relative}$)

	rs_1	rs_2	rs_3	rs_4	rs_5	rs_6	rs_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)

5.4 Information Distilling

We propose two policies, frequency-based and rarity-based, to distill information from different views. Rarity and frequency basically occupy two opposite 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 IV. 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. Note that for different datasets and diverse specified ego node, users can interact with our information abstraction system by controlling the parameters of step size k and the filtering threshold δ. These parameters allow informative flexibility and tolerate the cases sensitive to noise.

Table IV. The four abstraction measures from two viewpoints.

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

5.4.1 *Local Frequency*. It chooses the frequent $P(RS_1|x)$ relation sequences from the vectors as the important ones. For example, if the threshold δ is set to 2/7, only the top two frequent relation sequences in Table II (i.e., rs_6 and rs_7) are picked to represent x. In other words, rs_1 to rs_5 are filtered out since they do not occur as frequent as other relation sequences with respect to x. The idea behind this view is that x is summarized by the most frequent behaviors it involves.

5.4.2 *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 same example δ=2/7, rs_1 and rs_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 relation sequences whose conditional probabilities are 0 such as rs_3 and rs_4. The idea behind this view is that rare relation sequences could indicate

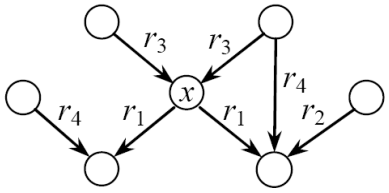
something that should not happen but in fact still occurs, and thus demands more attention. The other reason such a view of abstraction should exist is that rare events in a large network are generally harder to detect than frequent ones.

5.4.3 *Relative Frequency*. This uses Table III instead of Table II. $P(X_2=x|RS_2=rs)$ represents how frequently the ego x is involved in rs compared to other nodes. Since $\Sigma_X P(X_2|RS_2)=1$, we can treat each column in Table III as a relative comparison among all nodes for a certain relation sequence rs. Then $P(X_2=x|RS_2=rs)$ is representative of x if this value is relatively high compared to other nodes. In the example, rs_3 and rs_6 will be chosen to represent x since they are relatively high (i.e., ranked 2nd and 9th) 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 $P(X_2|RS_2)$. For instance, it might not make sense to compare the number of publications among people from different research areas.

5.5 Abstracted Graph Construction

Now we have distilled features as the abstraction for an ego node. One plausible form is to report distilled relation sequences and corresponding probabilities to the users. Though it seems to be a reasonable output since $P(RS_1|X_1)$ or $P(X_2|RS_2)$ 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 relation sequences and the corresponding nodes.

Figure 11 illustrates our idea. Assume we want to keep the top 2 ranked relation sequences and filter out the rest. The relation sequence of the highest score (e.g., rs_1) is first used to match the original network to obtain a subgraph that originates from the ego x and contains all the nodes involved in rs_1 (see Figure 12(a)). The same action is then performed on the next best relation sequence. The final abstracted graph of the ego node x is shown Figure 12(b).

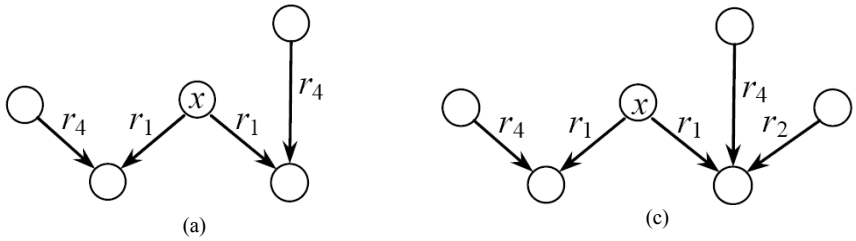


(a)

ID	Ranked Relation Sequence	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. 11. (a) An example $H_{k,x}$ and (b) the ranked relation sequences with scores.



(a) (c)

Fig. 12. (a) The abstracted graph after adding rs_1 (b) The final graph after rs_1 and rs_2 are added.

Note that it is not feasible to produce the abstracted graph by removing the discarded relation sequences since edges involved in one relation sequence might also occur in others. Therefore, eliminating one of them will sometimes cause the informative relation sequences to disappear. The complete algorithm of our egocentric information abstraction is given in algorithm 3. 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 2-8). According to the three designated viewpoint of abstraction, the most relevant relation sequences are picked (line 9-17). Finally, the abstracted graph is constructed incrementally (line 18-22).

Algorithm 3. Egocentric Information Abstraction

Input: H: a heterogeneous network; x: the query ego node; k: the step size for relation sequences; δ: the information filtering threshold; *view*: policy for information distilling
Output: H^{abs}: the abstracted graph from different views

1: Extract the k-step neighbor subgraph $H_{k,x}$ of x.
2: // derive the table of local measure
3: $tbl_{local} = P(RS_1|X_1)$ using SP.
4: // derive the table of relative measure and rank each column
5: $tbl_{relative} = P(X_2|RS_2)$ using SP.
6: **for** j = 1 to $|signatures|$ **do**
7: Compute the ranked value of $tbl_{relative}(:, j)$ in descending order.
8: **end for**
9: $distilledSet = \{\}$. // collect the signature of top score of specified view
10: **if** *view* = "localFrequency" **do**
11: $distilledSet = distilledSet \cap$ argmaxOfTop$\delta(tbl_{local}(x, signature))$.
12: **else if** *view* = "localRarity" **do**
13: // note that those scores equal to zero are ignored
14: $distilledSet = distilledSet \cap$ argminOfTop$\delta(tbl_{local}(x, signature))$.
15: **else if** *view* = "relativeFrequency" **do**
16: $distilledSet = distilledSet \cap$ argmaxOfTop$\delta(tbl_{relative}(x, signature))$.
17: **end if**
18: Let $H^{abs} = NULL$.
19: **for** $sig \in distilledSet$ **do**
20: $instances$ = Find path instances in $H_{k,x}$, whose relation sequence equals to sig.
21: $H^{abs} = H^{abs} \cap instances$.
22: **end for**
23: **return:** H^{abs}.

6. EXPERIMENTAL STUDIES

Evaluation is generally a challenging issue for social network analysis. Tasks such as centrality, clustering and abstraction naturally do not possess a unique and authoritative answer (therefore people have proposed different kinds of algorithms to tackle things from different aspects). Nevertheless, we argue that having no gold standard in nature should not become an original sin that hinders the progress of an area. Therefore we design several diverse experiments for our model and tasks using both artificial and natural datasets, and hopefully the evaluations from different angles can provide a more general explanation about what kind of outputs the proposed algorithms can produce as well as their value.

6.1 Experiment Design

The focus of our first evaluation, following a similar strategy as other centrality algorithms [14][17][47], is mainly about demonstrating what kind of results can be obtained from the three

proposed centrality methods as well as their meaning and uniqueness compared with the results acquired from other methods. The second experiment we conducted is to perform a role-based clustering on a movie dataset. This evaluation is not aiming to prove the correctness of the results (again, there is no apparent gold standard for roles in a heterogeneous network), instead it is designed to demonstrate the difference between our role-based clustering and the conventional community detection as well as provide intuitive insight to show the validity of our outputs. Borrowing from the conventional evaluation strategy utilized in the visualization society, our third experiment emphasizes displaying several abstracted social networks obtained from different egocentric views, and based on which one can easily grasp the meaning and usage scenario of the algorithm. The above three experiments are performed on top of a natural movie dataset acquired from the UCI KDD Movie repository. The fourth experiment is performed on a synthetic crime analysis task in which the gold standard is known. The goal is to find out whether egocentric abstraction can improve the accuracy and efficiency of human decisions in terms of identifying suspicious candidates from a heterogeneous social network.

6.2 Data Collection

In this section we elaborate how a natural and a synthetic heterogeneous social network are obtained for evaluation.

6.2.1 *UCI KDD Movie Dataset*. The first heterogeneous social network is generated from extracting entities and relations from UCI KDD Archive movie dataset [13]. 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 humans to analyze.

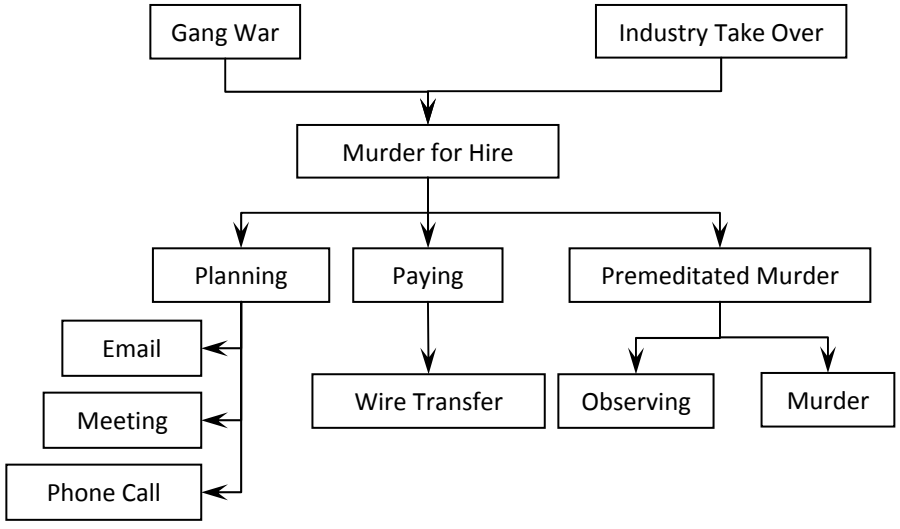


Fig. 13. Event-type hierarchy of the simulated Russian organized crime data.

6.2.2 *DARPA Crime Dataset*. The crime dataset we used is part of the simulated dataset developed during the US Defense Advanced Research Projects Agency (DARPA)'s Evidence Extraction and Link Discovery Program (see [31] for additional contexts). The data was generated by a simulator of a Russian organized crime (or Mafiya) domain that simulates the

entire process of ordering, planning, and executing high-level criminal activities such as murders for hire or gang wars. The hierarchy of event types is shown in Figure 13. 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 contains 9,429 nodes, and 16,257 links. There are 16 different node types (e.g., Murder, MurderForHire, Observing, and Planning) representing objects and events and 31 different link types (e.g., hasMember, eventOccursAt, victimintended, and socialParticipants) representing the relationships between those nodes. 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 occurs 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.

6.3 Results of Heterogeneous Centralities

In this experiment we apply our centrality measures to the same KDD movie dataset to identify central actors and compare the results with those acquired from some homogeneous centrality measures. Note that the homogeneous measures are computed by simply regarding all different relations as the same type. That is, to perform traditional centralities, all the typed information is removed from the heterogeneous network. The step size k for the modeling is set to 2. The results for homogeneous and heterogeneous centralities are shown in Table V and Table VI respectively, where we list the top-10 centered nodes and the corresponding scores based on each centrality.

We observe that nodes with high heterogeneous centrality scores are not necessarily the ones with high homogeneous centrality scores, and vice versa. Note that the person Hitchcock stands out in most of the cases because this dataset is collected and managed by a fan of Hitchcock. Therefore, there is more information about him than others and this consequently makes him a central one in most of the centrality analysis. One interesting observation is that Hitchcock is not a central node in our similarity-based centrality measure. This is because there are very few instances in this network that are similar to him. Therefore, he can seldom attract similar entities.

Here, we emphasize some actors to show the differences between the proposed centralities and the homogeneous ones. First, Jean Renoir and Anthony Quinn are both regarded as the central node in most of the heterogeneous centrality theories but none of the homogeneous ones. This indicates they are involved in *many kinds of relationships* with others, and they do not have sufficient connections with others nor sit in a crucial position to connect to other nodes to be detected by the homogeneous centrality methods. The real-world information tells us both Renoir and Quinn have several roles as an actor, director, writer, and producer. Therefore, it is reasonable to assume they have diverse connectivity with others through these multiple identities, and can be considered a legitimate central or important person in the movie society. The second interesting case is Humphrey Bogart. Differing from Renoir, he does not appear in the top-10 list of any heterogeneous centrality measure, but has a high degree centrality and eigenvector centrality. According to Wikipedia, he is one of the most popular actors of the early 20th century (has acted in 128 movies) and remained an actor for most of his life. This makes him a favorite node for homogeneous centrality but not as crucial in heterogeneous centrality measures.

Table V. The top-10 ranked results for homogeneous centrality measures.

	Homogeneous Centrality				Heterogeneous Centrality	
	Degree	Betweenness	Eigen.	Contribution	Diversity	Similarity
1	Alfred Hitchcock	Alfred Hitchcock	Alfred Hitchcock	Alfred Hitchcock	Alfred Hitchcock	Frank Sinatra
2	Martin Scorsese	Lyle R. Wheeler	Martin Scorsese	Martin Scorsese	Anthony Quinn	John Barrymore
3	Humphrey	Robert	Humphrey	Ernst	Ian	Julia

	Bogart	Goulet	Bogart	Lubitsch	McKellen	Roberts
4	Henry Fonda	Frances Fisher	Henry Fonda	Cecil B. DeMille	Spencer Tracy	Elia Kazan
5	Buster Keaton	Jennifer O'Neill	Burt Lancaster	Jean Renoir	Jean-Luc Godard	Judy Garland
6	Burt Lancaster	Wilford Brimley	Buster Keaton	John Wayne	Yves Montand	Anthony Quinn
7	James Stewart	Martin Scorsese	James Stewart	Elia Kazan	Maggie Smith	Buster Keaton
8	Gray Cooper	Derek Farr	Cary Grant	Anthony Quinn	Paul Papa	Jean Renoir
9	Cary Grant	Cecil B. DeMille	Gray Cooper	Sean Connery	Jeanette Macdonald	Robert Benchley
10	Vincent Price	Steven Spielberg	Vincent Price	Buster Keaton	Harry Ritz	Robert Duvall

Table VI. The top-10 ranked results for heterogeneous centrality measures.

	Heterogeneous Centrality					
	Contribution		Diversity		Similarity	
1	Alfred Hitchcock	28.41	Alfred Hitchcock	127	Frank Sinatra	136.50
2	Martin Scorsese	15.59	Anthony Quinn	70	John Barrymore	74.50
3	Ernst Lubitsch	13.01	Ian McKellen	69	Julia Roberts	73.75
4	Cecil B. DeMille	12.19	Spencer Tracy	66	Elia Kazan	70.00
5	Jean Renoir	12.00	Jean-Luc Godard	65	Judy Garland	69.25
6	John Wayne	11.06	Yves Montand	61	Anthony Quinn	68.50
7	Elia Kazan	11.01	Maggie Smith	60	Buster Keaton	67.75
8	Anthony Quinn	11.00	Paul Papa	59	Jean Renoir	66.00
9	Sean Connery	10.01	Jeanette Macdonald	58	Robert Benchley	65.00
10	Buster Keaton	10.00	Harry Ritz	57	Robert Duvall	63.25

This experiment on the real movie dataset shows that by taking advantage of higher-order relational information, our heterogeneous centralities can identify meaningful central individuals that can hardly be found by exiting homogeneous measures.

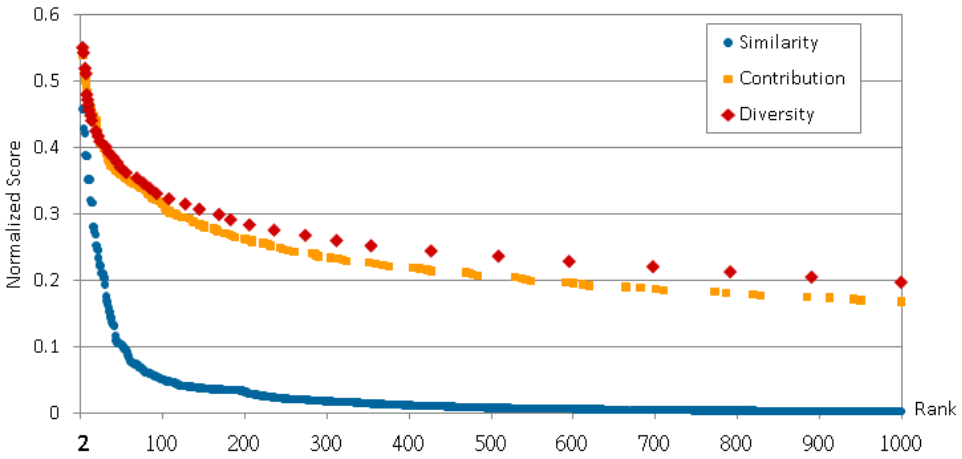


Fig. 14. Normalized heterogeneous centrality scores (diversity, contribution, and similarity) with respect to the top-1000 ranking list in the real movie dataset.

In addition, to realize and compare the three heterogeneous centralities, we also show the distributions of their centrality scores with respect to the top-1000 ranking lists on the real movie datasets, as given in Figure 14. All the centrality scores are normalized to [0, 1], where the normalized scores of rank-1 equals to 1.0 and do not show in the figure for easy exploration. We can observe that the diversity-based centrality has similar trend of distribution as the contribution-based one. It is because both are directly computed from their signature profiles. Besides, the contribution-based centrality is stricter than the diversity-based one since it further considers the relative importance of each signature. Hence, the trend of the contribution-based one drops a bit faster than the diversity-based one. Moreover, the similarity-based centrality exhibits more significant distinguish ability and indicate that only few individuals in a heterogeneous network can attract those with similar behaviors in their neighborhood. For all the three proposed centralities, we can obtain some significantly central individuals if the top-50 ones are returned.

6.4 Results of Role-based Clustering

In this experiment, we validate our role-based entity clustering through demonstrating how it finds entities of different roles and comparing its result with the traditional community detection method.

Again we conduct the experiment using the UCI KDD movie dataset. For visualization and explanation purposes, here we extract only a subgraph from the original network and use RAT^2 to produce the signature profiles. Also, we choose the nodes of type "actor" to perform the role-based clustering. To be more precise, we generate a distance matrix based on the signature profiles of all actors in the extracted graph and apply K-means clustering algorithm to them. The number of clusters is set to 8 for visualization purpose. Simultaneously, we use the method of removals of high-betweenness edges method [26] to perform community detection for comparison.

Figure 15 displays the resulted graph. The shaded regions of different sizes and colors are clusters identified by the community detection algorithm, where there are a total of seven communities in the graph. To facilitate visualization, diverse types of relations (i.e., edges) are colorized differently, and the names of the nodes were not labeled. This graph demonstrates eight different role-based clusters we have identified, namely "A", "B", "C", "D", "E", "F", "G", and "H". Unlabeled nodes are non-actors (e.g., movies). One advantage of our approach is that it can produce the representative relation sequences of each role, as listed in Table VII. Such information can essentially provide the explanation of the clustering results.

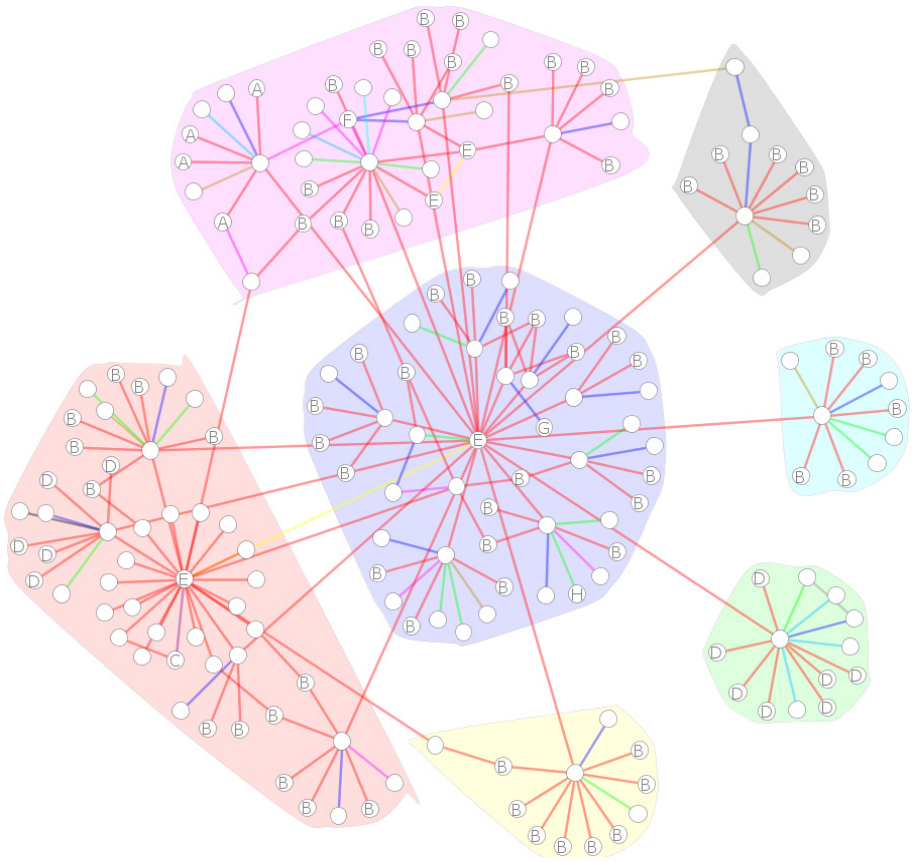


Fig. 15. The resulted graph of our role-based clustering.

Based on Figure 15, we can observe that the members in each role-cluster are scattered among different communities. It makes sense since actors in distinct groups could perform similar roles. We can further observe role "B" dominates a large portion of the network. Most of the relation sequences of role-B actors represent common behaviors for movie actors (e.g., such an actor played in a movie which has a certain director). These actors of role-B are generally regarded as full-time actors and seldom play other roles such as director, producer, or writer in their career. On the other hand, we can see the behaviors of those actors with role "A" and "D" differ from "B" to some extent. One special signature for actors of role "A" is they once played in some movie which was remade into other movies later. (e.g., Carrie Fisher and others performed in the movie "When Harry Met Sally", which was later remade into another movie "If Lucy Fell"). The actors of role "D" all cooperated with certain visual directors in a movie. The actors in role "C" (e.g., Randy Quaid) have some sibling(s) who also work in the movie industry (note that only movie-related persons are listed in this dataset). Role "E" includes people who married to another movie person, such as Dennis Quaid and Meg Ryan as well as Tom Hanks and Rita Wilson. Finally, classes "F", "G", and "H" represent people who play multiple roles such as actor-and-writer, actor-and-director, or actor-and-producer in this network. Based on k-means clustering, there happens to be only single instance satisfying certain criteria in this network. Therefore, they become single-element clusters (or outliers). Note that it is also possible to apply different kinds of clustering algorithms such as hierarchical clustering methods to generate different kinds of clustering results.

Table VII. The behaviors (i.e., relation sequences) that each kind of role involves in. Note the abbreviations: w="write script", m="musical direct", d="direct", p="produce", v="visual direct", and c="cinematize".

Role	Behaviors (i.e., relation sequences)
A	<hasActor^{-1}>, <hasActor^{-1}, w>, <hasActor^{-1}, m>, <hasActor^{-1}, remake^{-1}>, <hasActor^{-1}, d>, <hasActor^{-1}, hasActor>
B	<hasActor^{-1}>, < hasActor^{-1}, w>, <hasActor^{-1}, m>, <hasActor^{-1}, c>, <hasActor^{-1}, d>, <hasActor^{-1}, hasActor>, <hasActor^{-1}, p>, <hasActor^{-1}, remake>
C	<hasActor^{-1}>, <siblingOf>, <hasActor^{-1}, hasActor>, <siblingOf, spouseOf^{-1}>, <siblingOf, hasActor^{-1}>
D	<hasActor^{-1}, hasActor>, <hasActor^{-1}, m>, <hasActor^{-1}, v>, <hasActor^{-1}, p>, <hasActor^{-1}, d>, <hasActor^{-1}>
E	<hasActor^{-1}>, <spouseOf>, <spouseOf^{-1}>, <siblingOf^{-1}>, <-p>, <hasActor^{-1}, w>, <hasActor^{-1}, m>, <hasActor^{-1}, v>, <p^{-1}, d>, <hasActor^{-1}, remake^{-1}>, <hasActor^{-1}, c>, <hasActor^{-1}, d>, <spouseOf, siblingOf^{-1}>, <spouseOf^{-1}, p^{-1}>, <p^{-1}, hasActor>, <hasActor^{-1}, hasActor>, <siblingOf^{-1}, hasActor^{-1}>, <spouseOf^{-1}, hasActor^{-1}>, <hasActor^{-1}, p>, <hasActor^{-1}, remake>, <spouseOf, hasActor^{-1}>
F	<w^{-1}>, <d^{-1}>, <w^{-1}, remake>, <w^{-1}, w>, <w^{-1}, hasActor>, <w^{-1}, m>, <w^{-1}, c>, <d^{-1}, p>, <w^{-1}, p>, <w^{-1}, remake^{-1}>, <d^{-1}, hasActor>, <d^{-1}, remake>, <w^{-1}, d>
G	<d^{-1}>, <d^{-1}, hasActor>
H	<p^{-1}>, <p^{-1}, hasActor>, <p^{-1}, d>, <p^{-1}, w>, <p^{-1}, p>

6.5 Egocentric Abstraction on the Movie Dataset

We evaluate our egocentric abstraction model using the KDD movie dataset. First we choose "Meg Ryan", a famous actress, as the ego node to demonstrate the egocentric abstracted graphs. We have to point that this UCI KDD dataset is neither complete nor unbiased, therefore certain statistics collected based on it might not reflect the real-world status. The 2-step neighbor subgraph of "Meg Ryan" is shown in Figure 3 and the filtering threshold δ is arbitrarily set to 20%. Despite the seems-to-be small neighborhood size, we observe it is still quite complex (116 nodes, 137 edges and 18 different relation sequences).

The abstracted graph of local frequency is shown in Figure 16, capturing the regular behavior of Meg Ryan. We can observe she acted in many movies, especially in the comedic, dramatic, and romantic categories. Besides, her husband, Dennis Quaid, is also an actor in many movies. They co-starred in three of them.

The local rarity view is shown in Figure 17. It captures the rare behaviors. We can observe she is also a producer of a movie (i.e., the movie id is lak16). Besides, her husband's brother (i.e., Randy Quaid) also works in the movie industry (since only movie-related persons are listed in this dataset). Finally the movie (i.e., noe3) she acted in, the cinematographer (denoted as 'c') is listed in the dataset. This is a rare pattern for her because in her other movies the cinematographers are not listed.

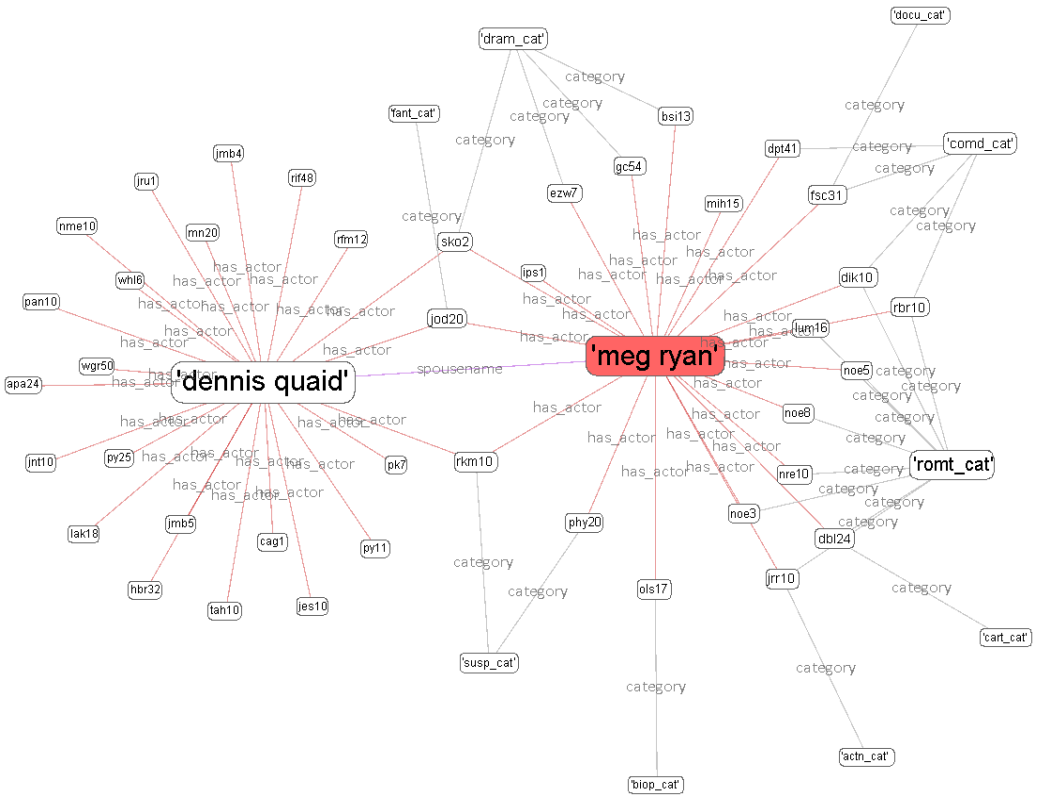


Fig. 16. Local frequency of "Meg Ryan."

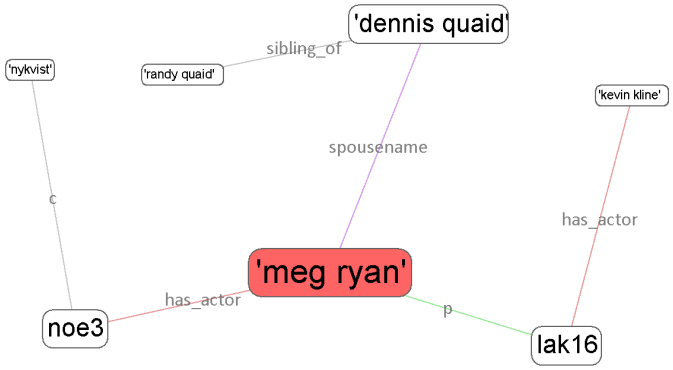


Fig. 17. Local rarity of "Meg Ryan."

The relative frequency view is shown in Figure 18, comparing the behaviors of Meg Ryan with other actors and showing she was significantly involved. We can see an interesting behavior in that she acted in relatively many remade movies. Also, she produced a movie (i.e., lak16), and such a behavior is not common for other actors. Finally, there is one rare path of her in the rarity view (i.e., her husband's sibling is also a movie person). This turns out to be rare among other actors, and thus becomes a relatively frequent behavior of hers (i.e., very few others in this dataset whose husband's sibling is also a movie person).

Fig. 18. Relative frequency of "Meg Ryan."

In this case study, we have used a heterogeneous movie network to demonstrate which kinds of information can be revealed through which egocentric 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.

6.6 Human Study for Crime Identification

In this experiment we evaluated the quality of our system through checking whether it can assist human subjects to identify the crime participants in an artificial dataset. We utilize the contract murder crime dataset as described in section 6.2. 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. Second, 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 about their decision given the abstracted information.

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 19 to 22, 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 relation sequences during abstraction. The black nodes are nodes representing criminal candidates.

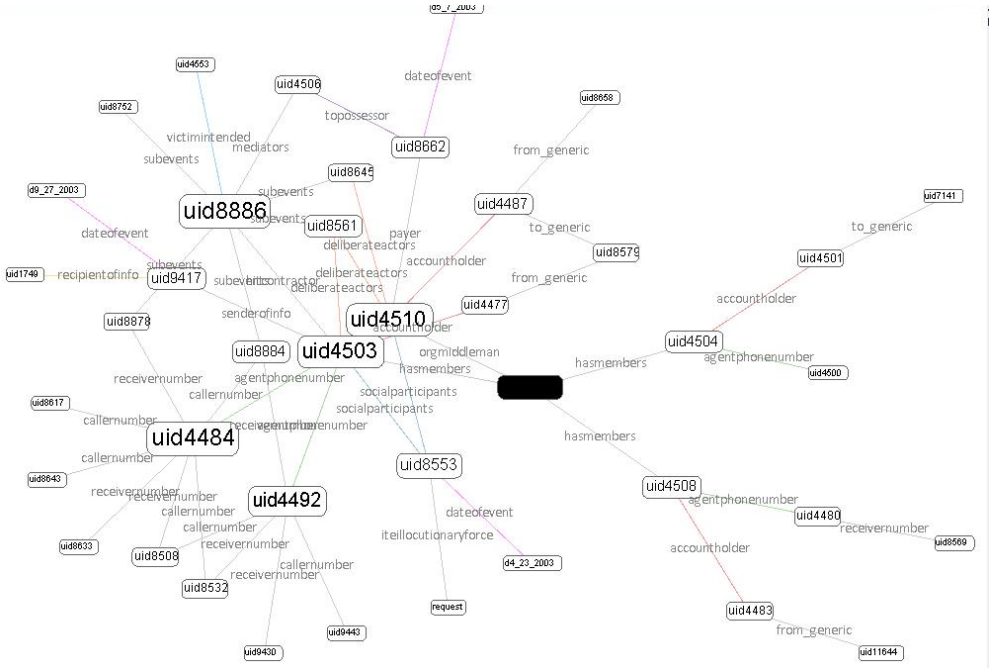


Fig. 19. The original 3-neighborhood graph.

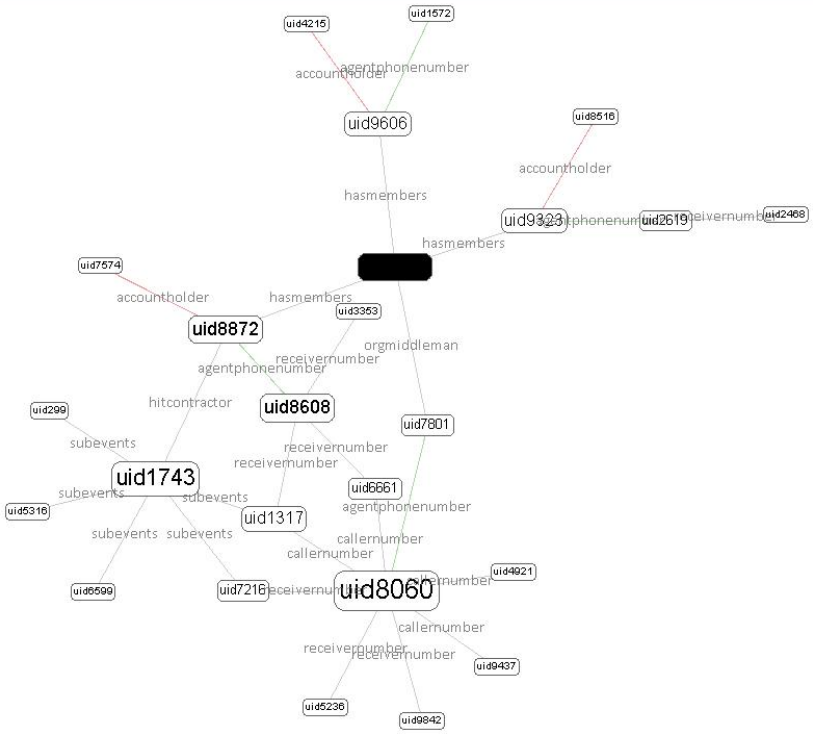


Fig. 20. Abstracted graph of local frequency.

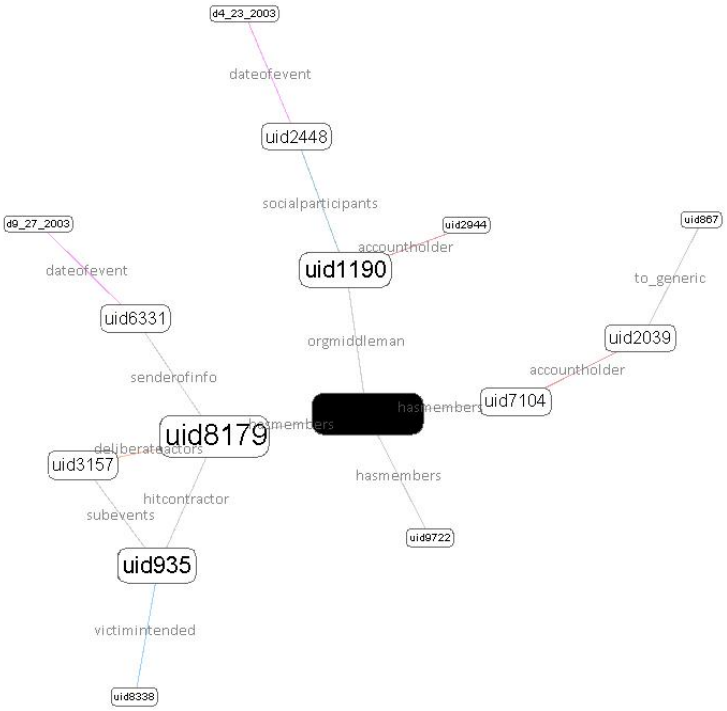


Fig. 21. Abstracted graph of local rarity.

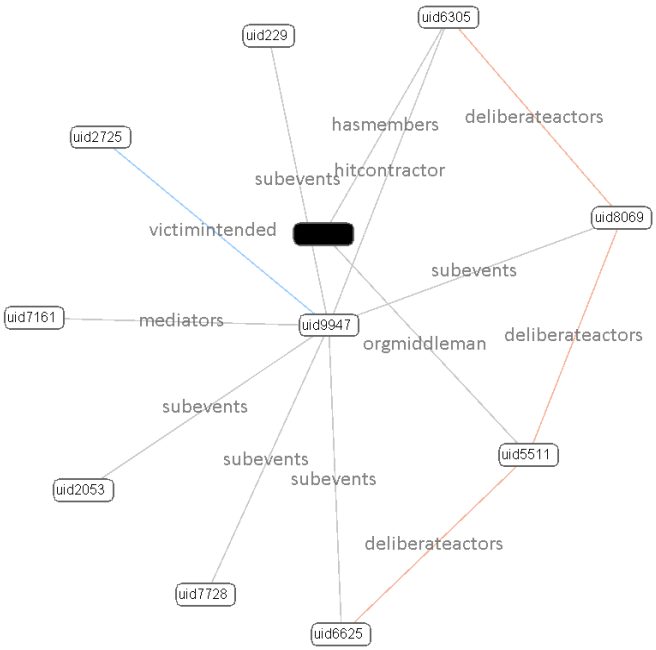


Fig. 22. Abstracted graph of relative frequency.

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

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

		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
	Local Rarity	44/60 (+8.7%)	13.9 ± 3.7	3.45 ± 0.33
	Relative Frequency	47/60 (+13.3%)	10.9 ± 2.2	3.73 ± 0.39

In terms of accuracy, the results show the users can at least do as well as using the original graph when using the abstracted ones. Since the non-abstracted graph contains the complete information, it makes sense to assume subjects can do as well 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 information are kept while some noise is filtered out, and therefore has less chance of misleading. The major improvement, as shown in the forth column of Table VII, lies in efficiency. The results show users utilize significantly less time (<50%) to reach better-quality results. The improved accuracy and efficiency truly demonstrate the abstraction can facilitate further human analysis since it retains critical information and significantly removes uninformative information.

In this dataset, there are some "key evidences" indicating the high-level events. After manually analyzing the three kinds of abstracted graphs, we have realized each abstraction view more or less captures different parts of those key evidences. For example, a kind of relation sequence representing "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 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.

6.7 Discussions

There are two issues worthy of further discussion.

- *Efficiency of the Proposed Algorithm*. There are two apparent bottlenecks for the proposed algorithms. The first is the generation of the relational tensor and the second is to estimate the conditional probabilities for abstraction. To generating the relational adjacency tensor efficiently, we are going to design a cloud computing platform using Hadoop/Map-Reduce framework that allows the computation in parallel. The second one lies in the need to sample a sufficient amount of representative paths for each the relation sequences. A technique called likelihood weighting, which has been applied successfully in the inference procedure of Bayesian Networks, can be applied here to force the occurrence of some rare events. Then the likelihood can be re-weighted based on the frequency of the forced decisions. This facilitates the design of an anytime algorithm. That is, we can still produce results of certain quality given insufficient time or resources, and the quality of the results can improve with increased time or resources.
- *Parameters*. Several parameters can be used to control the process of centrality, clustering, and abstraction: The first is the propagation distance k (or the k-step neighbor for the signature profile) and the second is the information filtering threshold for abstraction. Increasing k allows farer away nodes to come into play at the cost of efficiency and possibly introducing more noise; while increasing δ would boost the density of the abstracted graph. Given the small world phenomenon in most of the social networks, k shall be set to a small number. We

recommend determining k based on the connectivity of nodes in the network and δ according to the number of different relation types.

7. CONCLUSIONS AND FUTURE DIRECTIONS

In this research, we present a novel framework for knowledge discovery in heterogeneous social networks. Complex information about the graph topology and relational semantics is modeled through an unsupervised, automatic, and robust mechanism.

Here, we summarize the contribution in the following points:

- A tensor-based relational adjacency model with operations about relation sequences is proposed to catch the direct and indirect information for nodes. This model can simultaneously capture the topological and relational semantics of a heterogeneous network. Besides, this model is succinct yet powerful, and it is modularized enough to facilitate fast implementation.
- We define three brand-new centrality measures for heterogeneous social networks, including contribution-based, diversity-based and similarity-based centrality. And each estimates the nodes' importance from distinct points of view. The experiments on a real-world movie dataset demonstrate that it can truly identify central nodes that are otherwise hard to find using existing methods.
- We present a role-based clustering schema to group nodes based on their higher-order relational connections in the network. An experiment is conducted to explain its effectiveness and difference compared with the conventional community detection algorithm.
- We propose the ego-centric abstraction problem as well as its solutions. Three viewpoints, including local frequency, local rarity, and relative frequency, are provided to extract different aspects of important information from the network. We also propose an incremental method to reconstruct the abstracted graph for advanced exploration and visualization. The experiments are conducted on both real-world and synthetic dataset. The outcomes not only demonstrate the usability of our approach but also show the designed egocentric abstraction can assist human analyst in making more accurate, efficient, and confident decisions.

There are two future directions:

- *From Individual to Group*. So far we have only defined centrality at the individual level. For some real-world marketing applications, we think the group-centric importance measures can be significantly beneficial. Therefore, in the future we would like to extend our model to consider the centrality of a group of nodes.
- *From Static to Dynamic*. One important future plan is to extend the model of the relational adjacency tensor (RAT) to the time-evolved domain by adding the 4^{th} order element (i.e., time) so it is possible to pursue mining in dynamic heterogeneous social networks. For example, the evolution of individual's behavioral pattern can be formulated using time-based signature profiles.

REFERENCES

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

[2] P. Bonacich. 1972. Factoring and Weighting Approach to Status Scores and Clique Identification. *Journal of Mathematical Sociology*, 2(2), 113–120.

[3] M. Barthelemy, E. Chow, and T. Eliassi-Rad. 2005 Knowledge Representation Issues in Semantic Graphs for Relationship Detection. In *Proceedings of AAAI Spring Symposium on AI Technologies for Homeland Security (AAAI-SS'05)*, 91–98.

[4] R. Breiger, S. Boorman, and P. Arabie. 1975. An Algorithm for Clustering Relational Data with Application to Social Network Analysis and Comparison with Multidimensional Scaling. *Journal of Mathematical Psychology*, 12: 328–383.

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

[6] J. Chen, O.R. Zaiane, and R. Goebel. 2009. Detecting Communities in Social Networks Using Max-Min Modularity. In *Proceedings of SIAM International Conference on Data Mining (SDM'09)*, 978–989.

[7] N. Du, B. Wu, and B. Wang. 2007. Backbone Discovery in Social Networks. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, 100–103.

[8] M. Field. 1973. Algebraic Connectivity of Graphs. *Journal of Czechoslovak Math*, 23:298–305.

[9] L.C. Freeman. 1979. Centrality in Social Networks: Conceptual Clarification. *Social Networks*, 1(3), 215–319.

[10] L.C. Freeman. 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(6), 35–41.

[11] M.S. Granovetter. 1973. The Strength of Weak Ties. *American Journal of Sociology*, 78(6).

[12] R. Guimera and L.A.N. Amaral. 2005. Functional cartography of complex metabolic networks. *Nature* 433, 895–900.

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

[14] W. Hwang, T. Kim, M. Ramanathan, and A. Zhang. 2008. Bridging Centrality: Graph Mining from Element Level to Group Level. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, 336–344.

[15] B.W. Kernighan, and S. Lin. 1970. An Efficient Heuristic Procedure for Partitioning Graphs. In *Bell System Technical Journal*, 49:291–307.

[16] J. Kleinberg. 1999. Authoritative Sources in Hyperlinked Environment. *Journal of ACM*, 46(5), 604–632.

[17] V. Latora, and M. Marchiori. 2007. A Measure of Centrality Based on the Network Efficiency. *New Journal of Physics*, 9(6):188.

[18] C.T. Li, and S.D. Lin. 2009. Egocentric Information Abstraction for Heterogeneous Social Networks. In *Proceedings of International Conference on Advances in Social Network Analysis and Mining (ASONAM'09)*, 255–260.

[19] S.D. Lin. 2007. Modeling, Searching and Explaining Interesting Instances in Multi-Relational Network. PhD Dissertation, University of Southern California.

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

[21] Y.R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. 2009. MetaFac: Community Discovery via Relational Hypergraph Factorization. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, 527–535.

[22] B. Long, Z.M. Zhang, X. Wu, and P.S. Yu. 2006. Spectral Clustering for Multi-type Relational Data. In *Proceedings of International Conference on Machine Learning (ICML'06)*, 585–592.

[23] E. Minkov, and W.W. Cohen. 2007. Learning to Rank Typed Graph Walks: Local and Global Approaches. In *Proceedings of ACM SIGKDD Workshop on Web Mining (WebKDD) and Social Network Analysis (SNA-KDD)*, 1–8.

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

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

[26] M.E.J. Newman, and M. Girvan. 2004. Finding and Evaluating Community Structure in Networks. *Physics Review*, E 69.

[27] M.E.J. Newman. 2006. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Physics Review*, E 74.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. In Technical Report, Stanford University.

[29] D. Rogers, and M. Hahn. 2010. Extended-Connected Fingerprints. *Journal of Chemical Information and Modeling*, 50(5), 742–754.

[30] V. Satuluri, and S. Parthasarathy. 2009. Scalable Graph Clustering Using Stochastic Flows: Application to Community Detection. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, 737–745.

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

[32] J. Scripps, P.N. Tan, and A.H. Esfahanian. 2007. Exploration of Link Structure and Community-Based Node Roles in Network Analysis. In *Proceedings of IEEE International Conference on Data Mining (ICDM'07)*, 649–654.

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

[34] J. Shetty, and J. Adibi. 2004. Discovering Important Nodes through Graph Entropy: The Case of Enron Email Database. In *Proceedings of ACM Workshop on Link Discovery (LinkKDD'05)*, 74–81.

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

[36] J. Sun, D. Tao, and C. Faloutsos. 2006. Beyond Streams and Graphs: Dynamic Tensor Analysis. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, 374–383.

[37] Y. Sun, Y. Yu, and J. Han. 2009. Ranking-Based Clustering of Heterogeneous Information Networks with Star Network Schema, In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, 797–806.

[38] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. 2008. Community Evolution in Dynamic Multi-Mode Networks. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, 677–685.

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

[40] D. Vincent, and B. Cecile. 2005. Transitive Reduction for Social Network Analysis and Visualization. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 128–131.

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

[42] D.J. Watts, and S.H. Strogatz. 1998. Collective Dynamics of Small-world Networks. *Nature* 393, 440–442.

[43] H.C. White, S. Boorman, and R. Breiger. 1976. Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions. *American Journal of Sociology*, 81: 730–780.

[44] A.Y. Wu, M. Garland, and J. Han. 2004. Mining Scale-free Networks Using Geodesic Clustering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 719–724.

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

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

[47] D. Zhou, S.A. Orshanskiy, H. Zha, and C.L. Giles. 2007. Co-Ranking Authors and Documents in a Heterogeneous Network. In *Proceedings of IEEE International Conference on Data Mining (ICDM'07)*, 739–744.

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

[49] C.T. Li, and S.D. Lin. 2010. Mining Heterogeneous Social Networks for Egocentric Information Abstraction. In Book Chapter *"From Sociology to Computing in Social Networks"*, *Lecture Nodes in Social Networks*, Vol.1, 35–57.