Visual Mining of Multi-Modal Social Networks at Different Abstraction Levels

Lisa Singh
Georgetown University
Washington, DC
singh@cs.georgetown.edu

Mitchell Beard
Georgetown University
Washington, DC
msb4@georgetown.edu

Lise Getoor
University of Maryland
College Park
getoor@cs.umd.edu

M. Brian Blake
Georgetown University
Washington, DC
mb7@georgetown.edu

Abstract

Social networks continue to become more and more feature rich. Using local and global structural properties and descriptive attributes are necessary for more sophisticated social network analysis and support for visual mining tasks. While a number of visualization tools for social network applications have been developed, most of them are limited to uni-modal graph representations. Some of the tools support a wide range of visualization options, including interactive views. Others have better support for calculating structural graph properties such as the density of the graph or deploying traditional statistical social network analysis. We present Invenio, a new tool for visual mining of socials. Invenio integrates a wide range of interactive visualization options from Prefuse, with graph mining algorithm support from JUNG. While the integration expands the breadth of functionality within the core engine of the tool, our goal is to interactively explore multi-modal, multi-relational social networks. Invenio also supports construction of views using both database operations and basic graph mining operations.

1 Introduction

With the growth of large social communities on the Web and the use of more sophisticated data collection methods, social network data continue to become more feature rich. Traditionally, research has focused on *uni-modal* networks with a limited number of features or variables. The focus has been on the relationships between nodes. In the case of a uni-modal network, each node represents a single object type, typically an actor, and each edge represents a relationship between two actors in the network [16]. The focus of uni-modal analysis involves calculating local and global network statistics to help interpret network relationships or flows. With the changing structure and increased complexity of social network data, a need exists to support a more generic social network model containing multiple node types (multi-modal), multiple edge types (multi-relational), and multiple descriptive features (multi-featured) associated with each. We will refer to this general representation as a *multi-modal, multi-relational, multi-featured* network or M^*3. While a number of visualization tools have been developed to support social network analysis of uni-modal networks, limited support exists for M^*3 network analysis. Furthermore, the strength of most existing tools is the ability to quickly calculate local and global network statistics and simply display the results visually. Even though this is an important first step to visual mining, a need exists to constrain and transform visual representations of larger networks to allow for interactive exploration of social networks. Our goal is to design a tool that bridges the gap between current social network visualization and graph mining software to better address the complexities of M^*3 social networks and to support the goals of visual analytics.

In this paper, we present Invenio, a new visual mining tool for social network data mining. Invenio integrates a wide range of interactive visualization options from Prefuse, with graph mining algorithm support from JUNG. While the integration expands the breadth of functionality within the core engine of the tool, our goal is to interactively explore multi-modal social networks by creating views that support visual analytics using both database operations and basic graph mining operations. The views represent multiple abstraction levels of a single social network.

The paper is organized as follows. The next section overviews some social network concepts and discusses abstraction levels of multi-modal networks. Section 3 describes related tools that can be used for visualization and mining of social networks, identifying current capabilities and limitations. Section 4 describes Invenio, focusing on its functionality and various contributions, including analyzing data at multiple abstraction levels when applying graph mining operations. An example visualization analysis is presented in Section 5 on a real world social network. Finally, we conclude with future directions.

2 Social Network Data Representations

Throughout the section, we use terms from the field of social networks. For a more detailed discussion of social network theory, see Wasserman and Faust [16].

A social network contains people (actors) linked or connected by social relationships [13]. Examples of social networks include publication networks, corporate executive networks, disease transmission networks, sport team networks, family lineage networks, and online community networks. Many different types of relationships exist between actors in a network, including affiliation, kinship, interaction, flow of resources, and cognitive.

Definition 1 *An affiliation network N consists of a set of actors A, linked via a set of relationships R to a set of events E, where*

$$\begin{aligned} A &= \{a_1, \ldots, a_{n_A}\}, \\ E &= \{e_1, \ldots, e_{n_E}\}, \\ R &= \{r_{ij}, |\ \textit{actor } a_i \textit{ participates in event } e_j\}. \end{aligned}$$

We define n_A to be the number of actors in A, n_E to be the number of events in E, and n_R to be the number of relationships in R. In this context, R is equivalent to an associative mapping between actors in A and events in E.

Actors are the base element in the social network. We refer to a distinct group of actors as an *actor set*. Typically, actors are people. This is not a requirement, but we can assume so without loss of generality. We define an *event set E* to be any entity set or relation through which actors can be linked. Example event sets include corporations, publications, blogs, emails, and diseases. While some event sets are not literally 'events', we choose to keep our terminology consistent with that of traditional social network theory. In more general affiliation networks, many different actor sets, event sets, and relationship sets may exist in a single network. For example, there may be one actor set for customers and one for employees; one event set for stores and one for products; and multiple relationships between them, e.g. employed by, purchased by, returned to, etc.

Therefore, we extend the basic affiliation network model $N = \{A, E, R\}$ to incorporate multiple actor sets A_x, event sets E_y and relationship sets R_z:

$$\begin{aligned} \mathbf{A} &= \{A_1, \ldots, A_{n_{AS}}\}, \\ \mathbf{E} &= \{E_1, \ldots, E_{n_{ES}}\}, \\ \mathbf{R} &= \{R_1, \ldots, R_{n_{RS}}\} \end{aligned}$$

where n_{AS}, n_{ES}, n_{RS} are the number of actor sets, event sets, and relationship sets in N, respectively.

Definition 2 *We define the extended network $N = \{\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{R}\}$ as a multi-modal, multi-relational social network, where modal is associated with the number of node sets in N and relational refers to the number of relationship types in N. The base case remains a single mode and a single relationship type.*

A network containing a single actor set and no event sets is referred to as a uni-modal network. A network containing a single actor set and a single event set or two actor sets is referred to as a bi-modal network. As we will see in the next section, most visualization software supports analysis of uni-modal and bi-modal networks.

For each element in the network, descriptive attributes and structural features can be associated with it. *Descriptive attributes* are features of A_x, E_y or R_z that are specific to the social context of the network. A disease transmission social network may contain descriptive attributes representing the location of a person's home and the date of disease discovery. In contrast, a corporate board social network may contain descriptive attributes representing a job function and the tenure of a board member. While some descriptive attributes may be better represented as a relationship value between an event set and an actor set, keeping some attributes associated with a node can be important sematically for accurately constraining a network during visual mining.

In contrast, *structural properties* are features extracted from the network structure. Some structural properties are specific to individual nodes and edges in the network. An example of a structural property is *degree centrality*. Degree centrality $d(a_i)$ is the number of actor nodes adjacent to actor node a_i. Other well-known structural properties include closeness and betweenness. We refer the reader to [16] for more details about network properties. Attributes of network components, nodes or edges, may be structural, descriptive or both. A network supporting both attribute types is considered multi-featured.

It is useful to translate these abstract graph representations into a notation consistent with relational database theory. Therefore, we introduce the following relations:

- $A_x(Id_{A_x}, B_{x1}, \ldots, B_{xr})$ where $x = 1$ to n_{AS}

- $E_y(Id_{E_y}, C_{y1}, \ldots, C_{ys})$ where $y = 1$ to n_{ES}
- $R_z(Id_{A_x}, Id_{E_y}, D_1, \ldots, D_t)$ where $x = 1$ to n_{AS}, $y = 1$ to n_{ES}, and $z = 1$ to n_{RS}

The complete network N is defined by the full set of relations, $N = \{A_1, ..., A_{n_{AS}}, E_1, ..., E_{n_{ES}}, R_1, ..., R_{n_{RS}}\}$. Notice that relations R_z are the basis for determining the edges in the network. Here the (Id_{A_x}), (Id_{E_y}), and (Id_{A_x}, Id_{E_y}) are primary keys for the relations A_x, E_y and R_z, respectively. Figure 1 shows an example multi-modal network in this notation. In the example, there is one actor relation (*A*), two event relations (*E1*, *E2*), and two relationship relations (*R1*, *R2*). Each relation shows sample attribute values.

There are a number of advantages to this multi-modal interpretation of social networks. First, it allows us to represent a rich set of features for both nodes and edges. These features (descriptive attributes and structural properties) can be used to focus on subgraphs or prune the data set both for visualization clarity and prior to running an expensive graph mining algorithm for evaluation and prediction. For example, the user may choose to display nodes having a specific descriptive attribute value, thereby potentially reducing the number of relevant nodes and edges by one or more orders of magnitude. Second, this interpretation maps easily to the relational model in database theory [7]. This is important because the relational model supports more complex structure than traditional graph data sets and has an established set of relational operators that can be used to manipulate between graph topologies. Finally, it allows for easy translation between different topologies or abstraction levels. This enables users to see different representations or views of the data efficiently. In Section 4, we will see how Invenio uses the presented framework to accomplish this.

3 Social Network Visualization Tools

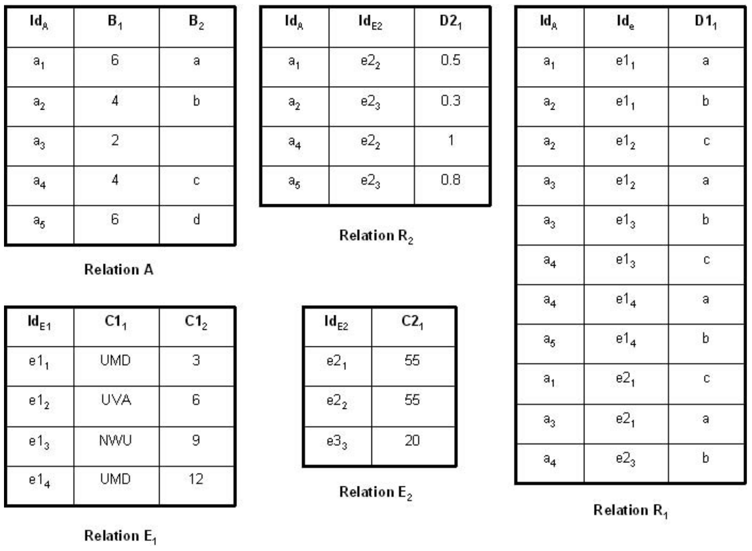
A number of surveys of social network tools exist [5], [6], [10], and [11]. We refer the reader to those for a more complete comparison of a larger number of tools. Here we focus our discussion on software packages that are well established or support visual mining objectives. This is not an exhaustive list, but rather a representative sample of the capabilities of current tools. While many of these tools have a wide range of features, to the best of our knowledge, none of them support visual exploration of M^*3 social networks. As we will see, some of them do support uni-modal, multi-relational network exploration. We begin by describing tools widely used by the social network community: Pajek [4], UCINet [2], and NetMiner [1].

Pajek is a non-commercial Windows based program for analyzing and visualizing larger uni-modal and bi-modal networks, in the tens to hundreds of thousands of nodes. The strength of this tool is its ability to support abstraction by recusively decomposing a large network into smaller networks that can be visualized more easily. However, the abstraction views are static, not interactive. Pajek also only supports single values for both nodes and edges. It does not support multi-featured nodes and edges. Pajek includes a large suite of statistical and mining algorithms using its matrix-based data structure. Algorithms include finding the k-core set, pattern matching, block modeling, maximum flow, shortest path, clustering, subnetwork extraction.

Similar to Pajek, UCINet supports uni-modal and bi-modal networks and contains matrix based algorithms for highlighting various groups and patterns in the network, including cohesion and centrality measures. Because of the underlying matrix representation of the graph, this commercial product supports merging and decomposing of matrices as well as translation of multi-valued attributes into binary variables for statistical operations. Each node can have multiple attribute values assigned by the user. A very nice feature is the ability to remove subsets of the graph based on values of node or edge structural properties. This allows for focused analysis of the network and supports visual data mining. Matrix operations can be conducted on larger networks as long as results are displayed as matrices and not graphs. UCINet does not contain visualization procedures, but outputs its results to Mage, NetDraw and Pajek for visualization. This lack of integration limits interactive visual exploration. NetDraw does have a larger degree of flexibility than Pajek for visually customizing the appearance of nodes and edges, as well as adding and removing them. Ego networks can also be focused on.

In contrast to UCINet, NetMiner is an integrated visual exploration tool for social network analysis. It accomplishes this by graphically presenting results of different statistical analyzes. For each node, a number of structural properties are pre-calculated. When one clicks on a node, its subnetwork is displayed and can be interactively manipulated. Each node can have multiple attribute values that are considered different relations. NetMiner allows for ego network analysis and subgraph extraction based on one or more attribute or affiliation values of a single node set. In general, this commercial product is an integrated visual exploration tool for uni-modal, multi-relational networks.

SocialAction is another new non-commercial software package in the visual mining arena [14] for large, multi-relational, uni-modal and bi-modal social networks. It lets a user select a ranking criterion of interest, e.g. betweenness, and ranks all the nodes and relationships using the criterion. From the ranked list, one visualizes an ego-network view of a selected node and sees the betweenness values to other nodes in the network. Nodes can also be filtered based on ranking. Users can also create a scatterplot between two separate rankings to determine the direction and strength of



Id_A	B_1	B_2
a_1	6	a
a_2	4	b
a_3	2	
a_4	4	c
a_5	6	d

Relation A

Id_A	Id_{E2}	$D2_1$
a_1	$e2_2$	0.5
a_2	$e2_3$	0.3
a_4	$e2_2$	1
a_5	$e2_3$	0.8

Relation R_2

Id_A	Id_e	$D1_1$
a_1	$e1_1$	a
a_2	$e1_1$	b
a_2	$e1_2$	c
a_3	$e1_2$	a
a_3	$e1_3$	b
a_4	$e1_3$	c
a_4	$e1_4$	a
a_5	$e1_4$	b
a_1	$e2_1$	c
a_3	$e2_1$	a
a_4	$e2_3$	b

Relation R_1

Id_{E1}	$C1_1$	$C1_2$
$e1_1$	UMD	3
$e1_2$	UVA	6
$e1_3$	NWU	9
$e1_4$	UMD	12

Relation E_1

Id_{E2}	$C2_1$
$e2_1$	55
$e2_2$	55
$e3_3$	20

Relation E_2

Figure 1. Example $M{*}3$ Network

a relationship between two features. Finally, nodes can be merged into cohesive subgroups based on link structure or attributes of the node.

The last group of software packages are graph toolkits that serve as frameworks for building social network and graph analysis tools. Prefuse [9], JUNG [12], and GUESS [3] fall into this category. Prefuse is a framework for developing dynamic visualizations of data. It uses a Java 2D graphics library and is scalable to having thousands of nodes displayed on the screen and millions of rows stored in a visualization table in memory. While the visualization is strong, the algorithmic and statistical support is limited. The developers of Prefuse used the toolkit to create Vizster, a visualization tool specifically for online social networks[8]. This software shows an ego-centric view of a person's social network. In this uni-modal network, each node has attributes that can be used to simplify or focus the display.

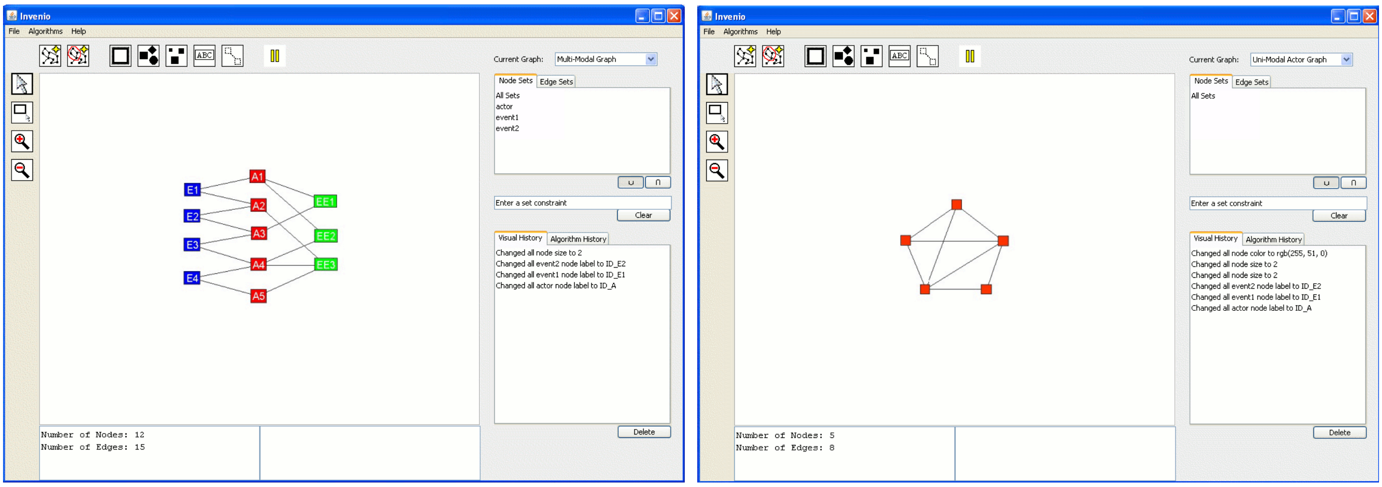
In contrast, JUNG is a java library (API) that has basic visual support for networks, but is designed to support a variety of representations of entities and their relations, including multi-modal graphs. It provides a mechanism for annotating graphs, entities, and relations with metadata. This toolkit facilitates the creation of analytic tools for complex data sets since it can examine the relationships between entities as well as the metadata attached to each entity and relationship. JUNG includes implementations of a number of algorithms from graph theory, data mining, and social network analysis, including clustering, decomposition, optimization, random graph generation, statistical analysis, and calculation of network distances, flows, and network importance measures (centrality, PageRank, HITS, etc.).

Finally, Guess is a language that extends the Jython system and combines analysis and visualization of graphs. It is a high level language toolkit (unlike Prefuse and JUNG) that lets users control visualization and prototype visualization applications. The strength of GUESS is the strong connection between the interpreted language and the visualization. It also has good algorithm support developed from the JUNG library and stores results in a backend database.

4 Invenio Visual Mining Software

This section overviews the design and functionality of the Invenio visual mining tool for social networks. Throughout, we describe some of the new features with a visual example of the relations presented in Figure 1. In the following section, we will use a real world data set to demonstrate results of a visual exploration task. Please note that the color versions of the figures are more informative than the black and white versions.

Figure 2(a) shows an example of the $M{*}3$ network using the Invenio software based on the relations presented in Figure 1. The center column of nodes contains the actors in the network. The left column of nodes contains events in E1 and the right column contains events in E2. This is the multi-modal view of the data as specified on the 'Current Graph' pull down at the top right of the screen. The user can also click on a node to display node attributes and structural properties.



Current Graph: Multi-Modal Graph

All Sets
actor
event1
event2

Changed all node size to 2
Changed all event2 node label to ID_E2
Changed all event1 node label to ID_E1
Changed all actor node label to ID_A

```
Number of Nodes: 12
Number of Edges: 15
```

Current Graph: Uni-Modal Actor Graph

All Sets

Changed all node color to rgb(255, 51, 0)
Changed all node size to 2
Changed all node size to 2
Changed all event2 node label to ID_E2
Changed all event1 node label to ID_E1
Changed all actor node label to ID_A

```
Number of Nodes: 5
Number of Edges: 8
```

Figure 2. Example network in Invenio (left: M^*3 representation)(right: Co-membership representation)

4.1 Network transformations

The other options in the pulldown menu on the top right include different projections or graph representations of the M^*3 network. There are times when we want to view the entire network while highlighting a particular node set or edge set. These translations are important because they can be used to help the user navigate through the graph. We now describe the creation of a uni-modal network from our multi-modal network.

To focus on actors, one can perform a unipartite projection of the actors in the multi-modal network. If we need to maintain details about relationship types in the actor projection, we can create a uni-modal, multi-relational network that has a single object type and multiple edge types. This uni-modal network is referred to as a *co-membership* graph; here the emphasis is on the actors. The objects in the co-membership graph are simply the actors in relation A_x and the edges can be constructed using the following algebraic expression to find instances of related actors:[1]

$$R_{CM(A_x)}(S.Id_{A_x}, R.Id_{A_x}) = \bigcup_{j=1}^{n_{RS}} \pi_{S.Id_{A_x}, R_j.Id_{A_x}} (\sigma_{S.Id_{A_x} < R_j.Id_{A_x}} (\rho_S(R_j) \underset{S.Id_E = R_j.Id_E}{\bowtie} R_j))$$

The R_{CM} relation describes an edge created for each pair of actors $S.Id_A$ and $R.Id_A$ which participate together in at least one event. The complete co-membership network is then defined as $N_{CM} = \{A, R_{CM}\}$ Notice that the co-membership graph loses information—we no longer know how many events the actors have in common. However, the CM relation is the foundation for the basic uni-modal social network used by many researchers the social sciences. Figure 2(b) shows the co-membership graph for our example.

Similarly, to focus on events of a single event type, one can perform a unipartite projection of an event set on R_z. This results in what is called an *event overlap* graph. It also contains a single node type and one or more edge types. In the event overlap graph, the emphasis is on the connections among events in one or more event sets. This graph has a node for each event, and an edge between events that share a common actor. The objects in the event overlap graph are also the events in relation E_y and we can construct the edges in the event overlap graph using the following algebraic expression to find instances of related events:

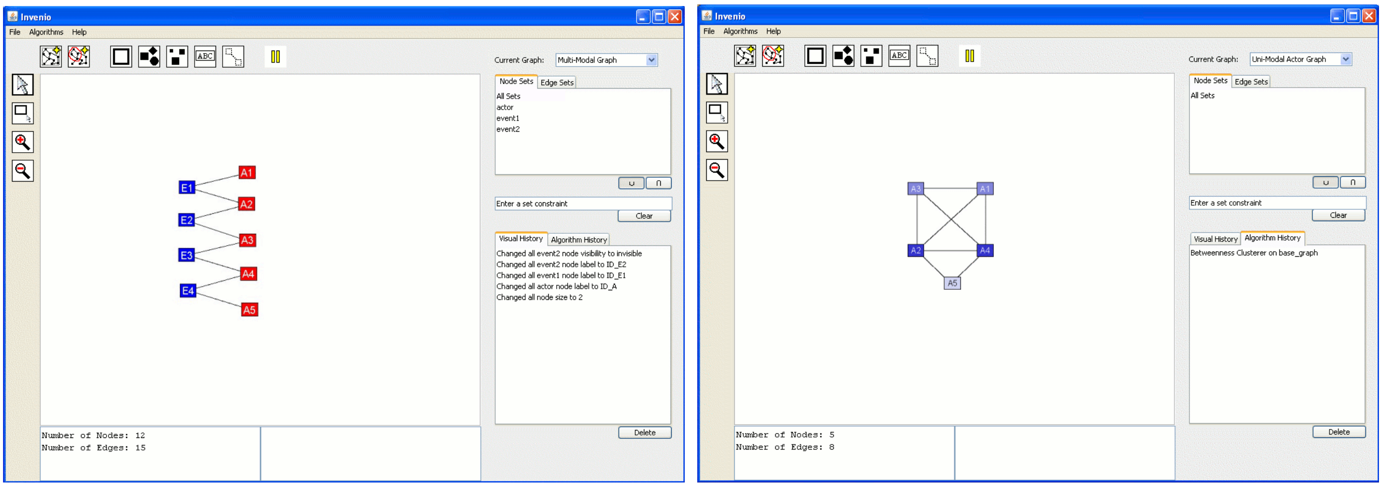
$$R_{EO}(S.Id_{E_y}, R_z.Id_{E_y}) = \bigcup_{j=1}^{n_{RS}} \pi_{S.Id_{E_y}, R_z.Id_{E_y}} (\sigma_{S.Id_{E_y} < R_z.Id_{E_y}} (\rho_S(R_z) \underset{S.Id_{A_x} = R_j.Id_{A_x}}{\bowtie} R_z))$$

Invenio supports visualization of event-overlap networks, as well as other uni-modal and multi-modal graphs.

4.2 Visual features

As previously mentioned, Invenio is a JAVA application that was developed using Prefuse and JUNG libraries.

[1] Recall that π, σ, $\bowtie$, and ρ are the relational projection, selection, join, and rename operators, respectively [7].



Current Graph: Multi-Modal Graph

All Sets
actor
event1
event2

Enter a set constraint

Changed all event2 node visibility to invisible
Changed all event2 node label to ID_E2
Changed all event1 node label to ID_E1
Changed all actor node label to ID_A
Changed all node size to 2

```
Number of Nodes: 12
Number of Edges: 15
```

Current Graph: Uni-Modal Actor Graph

All Sets

Enter a set constraint

Betweenness Clusterer on base_graph

```
Number of Nodes: 5
Number of Edges: 8
```

Figure 3. Example visual mining (left: subgraph creation)(right: clustering)

While we developed our own interface, we use a number of nice visualization features from Prefuse including libraries that change the appearance of nodes and edges, support for different graph layouts, e.g. spring embedder, animation actions, and basic expression support. The buttons on the left side of the top panel support options that allow the user to change the visual properties of a node set or edge set. Along with the basic visual options, e.g. color, shape, size, we also include options for labelling nodes based on any descriptive attribute or structural property of the nodes or edges, and information hiding at the node, edge, node group or entity level. This allows for focused subgraph creation based on relations. For example, suppose we are interested in creating a subgraph that focuses on actors related via relationship R1 to event E1. Then Invenio removes the relationship set $R2$ and the node set $E2$ from the visualization as shown in Figure 3(a). This subgraph could be easily added to the current graph pulldown by clicking on a button on the top panel.

4.3 Attribute Guided Subgraph Generation

During implementation, we extended the base data structures to allow for $M*3$ data sets. We then developed an environment for creating focused subgraphs based on attribute values. To generate attribute guided subgraphs, users can remove a subset of actors, events, and/or relationship attributes, thereby creating pruned social network $N' = \{A', E', R'\}$ [15]. We can prune actors by constraining or limiting the set of attribute values for attributes $B_1 ... B_r$.

$$A' = \cap_{i=1}^{r} \sigma_{A.B_i=b_i}(A)$$

For our example, we may choose to analyze only actor nodes (A) that have $A.B1 >= 4$. Similarly, we can prune events by constraining a set of attribute values of attributes $C_1 ... C_s$ and relationships or edges by constraining a set of attribute values of attributes $D_1 ... D_t$.

$$E' = \cap_{i=1}^{s} \sigma_{E.C_i=c_i}(E)$$

$$R' = \cap_{i=1}^{t} \sigma_{R.D_i=d_i}(R)$$

The support for relational constraints is on the right panel of the Invenio interface.

4.4 User Action History

Another feature on the right panel is the display of the history of user operations. Because this type of analysis is exploratory, recording the user actions can help streamline visual exploration. A user can choose to undo a set of operations to return to an earlier state of the analysis, or the user may want to keep track of the set of operations that helped generate a particular visual display to share with colleagues. The history gives users the ability to reproduce visual mining results. For clarity, visual history and algorithm history are displayed using two different tabs. Users also have the ability to store the state of the graphs they make during exploration.

4.5 Graph Mining

We also built a bridge between Prefuse and JUNG so that the suite of algorithms in JUNG could be used in conjunction with the Prefuse visualization libraries. So far we have only incorporated structural properties degree and betweenness, two clustering algorithms, and a shortest path algorithm. In the future, we will also include algorithms for interesting paths, statistical calculations, and other graph mining operations. These algorithms can be applied in Invenio from the *Algorithm* menu option. Figure 3(b) shows clustering results using the betweenness clustering algorithm on the actor uni-modal visualization. As a proof of concept, currently, users can choose to cluster the data using K-Mean or the Betweenness meaure. There are a few unique features of our interactive mining. First, the algorithm does not need to be applied to all the nodes. Any subset can be selected. Next, once applied, it can be reapplied as many times as necessary. For example, if we clustered a large graph, after using one clustering algorithm for a high-level clustering, another clustering algoritm can be applied to one or more of the clusters created. This interactive visual mining provides a systematic way to explore networks and reevaluate visualizations at any point.

4.6 Data Importing

Most social network tools read a specially formatted input file in order to generate visualizations. Invenio supports the loading of data directly from a relational database. The user provided a server location and password. Invenio reads all the relations in the database and lets the user select the relations of interest for visual analysis. The user can select any number of relations for nodes and edges. We also support subtable querying that allows only a targetted set of nodes to be imported for analysis.

5 Visualization Application Analysis

We now present a small exploratory analysis using Invenio. We created a data set containing the 2005 SIGMOD / PODS organizing and program committees, as well as the 2005 editorial boards of the VLDB Journal and the IEEE Transactions on Knowledge and Data Engineering. We used 3 modes (person, affiliation, and conference / journal) and relationships between each pair of node sets in the network. This data set has approximately 350 actors and 900 edges between the actors and the events.

Suppose we are interested in finding the actors that have participated on the most committees. Figure 4(a) shows the co-membership graph using a spring-force layout. Because some of the committees are large, the degree of each actor is also large, in the 100s. Therefore, the co-membership graph is dense with tens of thousands of edges. Since the number of events is much smaller than the number of actors, the multi-modal graph has fewer edges as illustrated in Figure 4(b). We can see the data clusters around four points; these are the journals and conferences. Because our example is interested in the relationship between the actors and the conferences / journals, we can remove the affiliation mode of the network to create a bi-modal graph (Figure 4(c)) and display only the actors with a degree greater than two (Figure 4(d)). The network is now constrained enough to analyze in detail. Further, recreating the subgraph is a repeatable procedure using the history information. If we want to change the final attribute constraint, we can undo the last operation and then specify a new constraint.

6 Conclusions and Future Directions

In this paper we presented Invenio, a new tool for visual mining and systematic analysis of M^*3 networks. We showed multiple approaches for subgraph creation based on the semantics of the data set. The importance of translating among different graph topologies cannot be overstated. It can significantly decrease the computational complexity of data mining algorithms and vastly improve the level of knowledge gained from visualization of social network data.

While Invenio has a number of new features, there are still enhancements that are needed. First, we plan to add more known algorithms and structural properties to the tool. Next, we want to enhance the visual flexibility of the tool by incorporating customizable panels, graph brushing and new layouts. Finally, while we have worked with social networks containing up to 1000 nodes, we still need to investigate extension for large graphs. Larger graphs can compromise performance and are difficult to view. Different clustering techniques and block modeling approaches have been introduced for uni-modal networks. We need to investigate extensions to them for M^*3 networks.

The synergy between the visual graphical models, the relational algebra framework / relational database back-end, the social network analysis, and the data mining algorithms allows for a more complete approach to analyzing M^*3 social networks. Invenio is a meaningful step toward this goal.

References

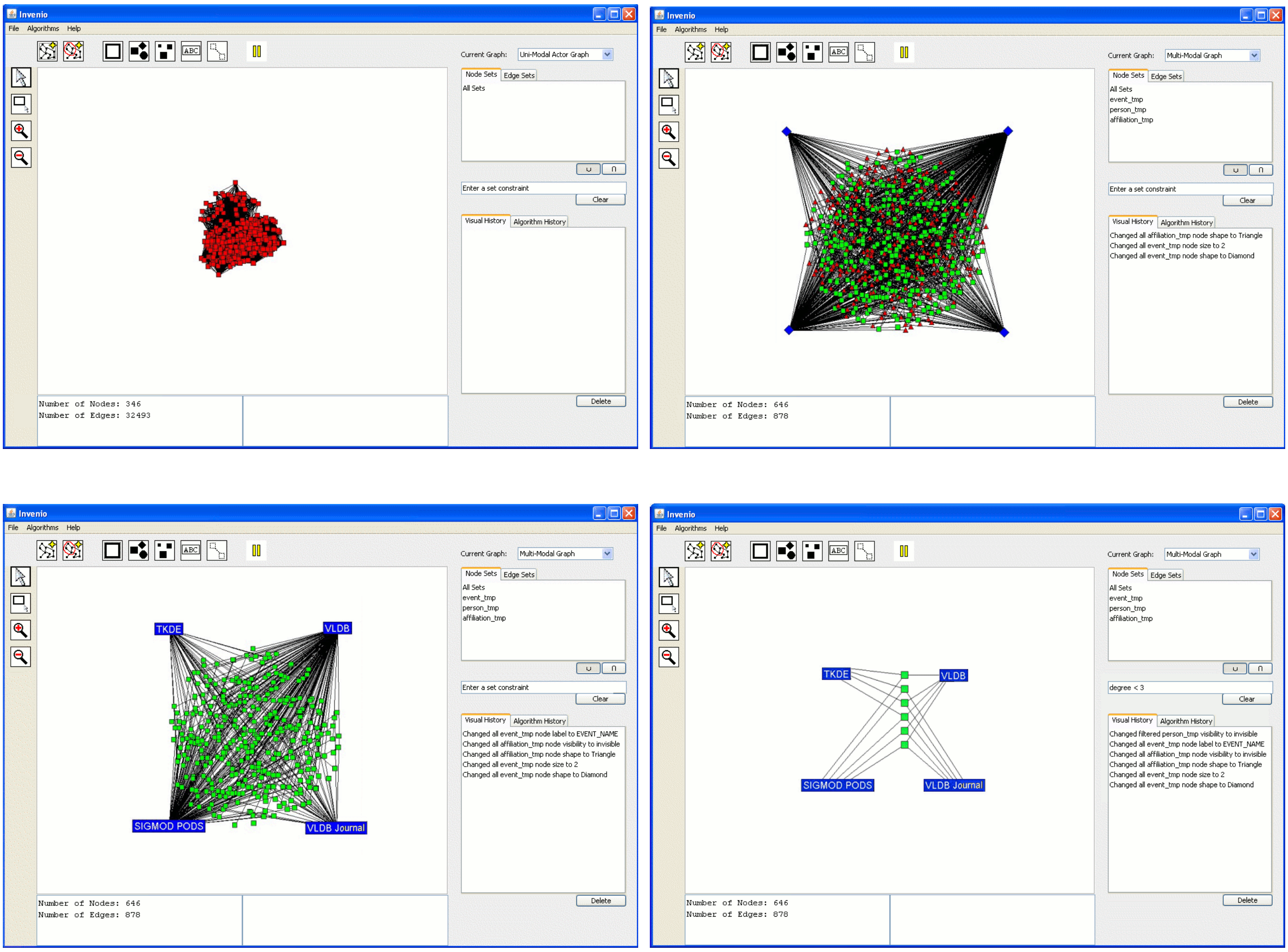
[1] Netminer ii: Social network mining software. website: http://www.netminer.com/NetMiner/home_01.jsp.

[2] Ucinet social network analysis software. website: http://www.analytictech.com/ucinet/ucinet.htm.

[3] E. Adar. Guess: a language and interface for graph exploration. In *SIGCHI conference on Human Factors in computing systems*, pages 791–800, New York, NY, USA, 2006. ACM Press.

[4] V. Batagelj and A. Mrvar. Pajek – program for large network analysis. *Connections*, 21:47–57, 1998.

[5] L. C. Freeman. Visualizing social networks. *Journal of Social Structure*, 1(1), 2000.



Current Graph: Uni-Modal Actor Graph

All Sets

Enter a set constraint

```
Number of Nodes: 346
Number of Edges: 32493
```

Current Graph: Multi-Modal Graph

All Sets
event_tmp
person_tmp
affiliation_tmp

Enter a set constraint

Changed all affiliation_tmp node shape to Triangle
Changed all event_tmp node size to 2
Changed all event_tmp node shape to Diamond

```
Number of Nodes: 646
Number of Edges: 878
```

Current Graph: Multi-Modal Graph

All Sets
event_tmp
person_tmp
affiliation_tmp

Enter a set constraint

Changed all event_tmp node label to EVENT_NAME
Changed all affiliation_tmp node visibility to invisible
Changed all affiliation_tmp node shape to Triangle
Changed all event_tmp node size to 2
Changed all event_tmp node shape to Diamond

```
Number of Nodes: 646
Number of Edges: 878
```

Current Graph: Multi-Modal Graph

All Sets
event_tmp
person_tmp
affiliation_tmp

degree < 3

Changed filtered person_tmp visibility to invisible
Changed all event_tmp node label to EVENT_NAME
Changed all affiliation_tmp node visibility to invisible
Changed all affiliation_tmp node shape to Triangle
Changed all event_tmp node size to 2
Changed all event_tmp node shape to Diamond

```
Number of Nodes: 646
Number of Edges: 878
```

Figure 4. Committee Member Data Set Visual Analysis (left: co-membership)(right: multi-modal)

[6] L. C. Freeman. Graphical techniques for exploring social network data. In J. Scott and S. Wassermann, editors, *Models and methods in social network analysis*. Cambridge University Press, 2004.

[7] H. Garcia-Molina, J. Ullman, and J. Widom. *Database Systems*. Prentice Hall, New Jersey, 2002.

[8] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *IEEE Symposium on Information Visualization (INFOVIS'05)*, page 5, Washington, DC, USA, 2005. IEEE Computer Society.

[9] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *SIGCHI conference on Human factors in computing systems*, pages 421–430, New York, NY, USA, 2005. ACM Press.

[10] I. Herman, G. Melan&#231;on, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[11] M. Huisman and M. A. J. van Duijn. Software for statistical analysis of social networks. In *International Conference on Logic*, 2004.

[12] P. S. S. W. J. O'Madadhain, D. Fisher and Y. B. Boey. Analysis and visualization of network data using jung. *Journal of Statistical Software*, 2005.

[13] E. O. Laumann, L. Galskeiwicz, and P. Marsden. Community structure as interorganizational linkages. *Annual Review of Sociology*, 4:455–484, 1978.

[14] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):693–700, 2006.

[15] L. Singh, L. Getoor, and L. Licamele. Pruning social networks using structural properties and descriptive attributes. In *International Conference on Data Mining*, 2005.

[16] S. Wasserman and K. Faust. *Social network analysis: methods and applications*. Cambridge University Press, Cambridge, 1994.