g-Miner: Interactive Visual Group Mining on Multivariate Graphs

Nan Cao
IBM Research
nan.cao@gmail.com

Yu-Ru Lin
University of Pittsburgh
yurulin@pitt.edu

Liangyue Li
HangHang Tong
Arizona State University
{liangyue,hanghang.tong}@asu.edu

ABSTRACT

With the rapid growth of rich network data available through various sources such as social media and digital archives, there is a growing interest in more powerful network visual analysis tools and methods. The rich information about the network nodes and links can be represented as multivariate graphs, in which the nodes are accompanied with attributes to represent the properties of individual nodes. An important task often encountered in multivariate network analysis is to uncover link structure with groups, e.g., to understand why a person fits a specific job or certain role in a social group well. The task usually involves complex considerations including specific requirement of node attributes and link structure, and hence a fully automatic solution is typically not satisfactory. In this work, we identify the design challenges for mining groups with complex criteria and present an interactive system, "g-Miner," that enables visual mining of groups on multivariate graph data. We demonstrate the effectiveness of our system through case study and in-depth expert interviews. This work contributes to understanding the design of systems for leveraging users' knowledge progressively with algorithmic capacity for tackling massive heterogeneous information.

Author Keywords

Group Mining; Visual Analysis; Information Visualization

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

The visual analysis of networks is an important subject in many domains ranging from social or information networks to power grids or biological networks. With the rapid growth of data available through various sources such as social media and digital archives, the interest in more powerful network visual analysis tools and methods is growing as well. These data often include rich information about the nodes and links and can be represented as *multivariate graphs*, in which the nodes are accompanied with attributes to represent the properties of individual nodes – e.g., the demographic information such as race or gender of the 'people' nodes in social network analysis. Such multivariate graphs provide rich contexts that enable analysts to form hypotheses while exploring the network [30]. An important task often encountered in their analysis is to uncover link structure with groups. For example, in social network analysis, the analysts are interested in understanding why a person fits a specific job or certain role well in a social group, comparing member composition across different groups, or looking for a group with specific link structure or member composition. Finding these groups allows them to determine common or irreplaceable parts of these networks which is for instance of importance for the maintenance in communication networks or discovering specific roles in organization networks.

Group mining with large-scale, heterogeneous network data is computationally intensive, and hence automatic group identification algorithms have become a popular solution. Recent advances in graph mining techniques, particularly in community detection [13] and team formation [1, 19] algorithms, have been developed to identify subsets from multivariate graph data. However, the automatically generated solutions derived from these techniques are rarely able to satisfy the complex or dynamic criteria that users require for finding desirable groups. More importantly, a fully-automatic approach can never support users' exploratory need, neither does it allow users to evaluate, refine or make sense of different solutions.

This paper presents an interactive system called g-Miner (Fig. 1) that enables visual analysis of groups with large-scale multivariate network data. Our key contributions include:

System. We introduce the problems of *interactive group mining* – concerning mining groups from multivariate network data with the capability to adapt complex or dynamic requirements of the desired group. We identify the system design requirements through a pilot user study. Guided by the design requirements, we integrate a set of graph mining algorithms with novel visualization tools to support iterative group mining with users' feedback in the loop.

Visualization. We proposed two sets of visualization tools for efficiently *locating* and *comparing* groups. (1) Cross-level exploration: The "Hierarchy Explorer" (Fig. 1(4)) allows users to navigate the entire graph dataset following a hierarchical structure. Once a group of interest is located, users can inspect the group via "Group Explorer" (Fig. 1(3)) or find alternatives of the group members via "Candidate Explorer"

CHI 2015, April 18–23 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04...$15.00.
http://dx.doi.org/10.1145/2702123.2702476

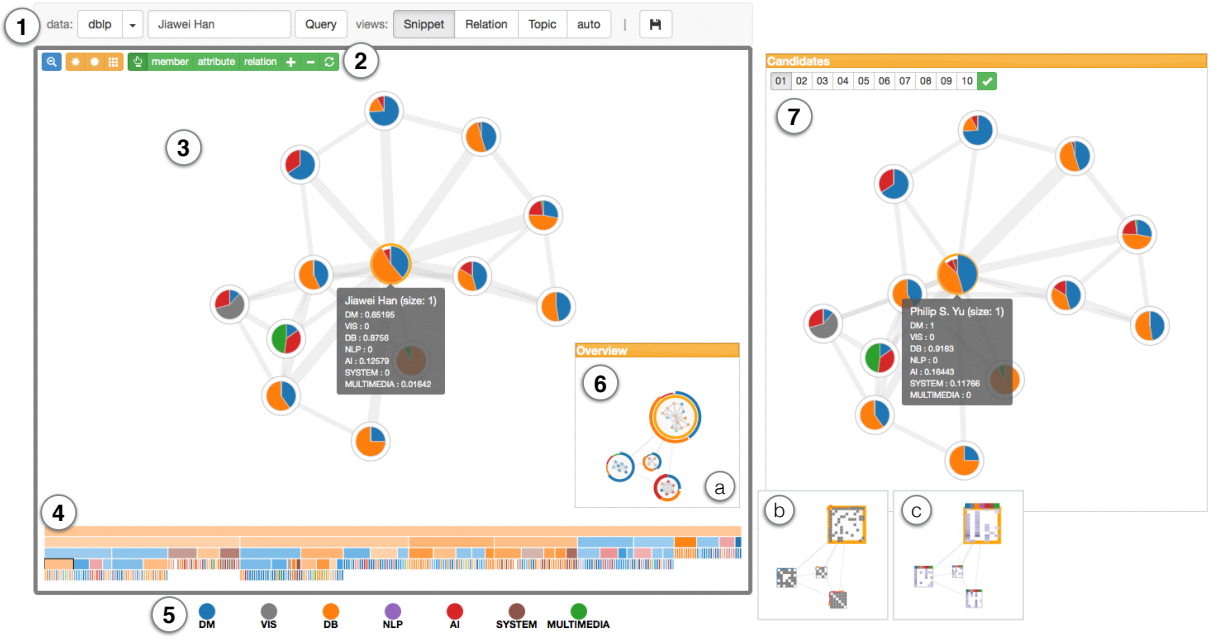

Figure 1. The visualization illustrates an example of using our proposed g-Miner system to find a candidate (e.g., Philip S. Yu) (in 7: Candidate Explorer) to replace the person (Jiawei Han) in the center of the network (in 3: Group Explorer) when that person is unavailable. Users can visually compare the two groups in terms of attribute and relational aspects and observe the candidate has similar expertise and social ties with the rest of the team. The labels from 1 to 7 correspond to the key interaction and visualization components detailed in the paper.

(Fig. 1(7)). (2) Multi-structure group views: Three different glyph-based views are automatically determined based on the topological and attribute structure of the group, including "Graph Snippet" (Fig. 1(a)), "Relation Map" (Fig. 1(b)) and "Feature Map" (Fig. 1(c)). These snippets highlight the relational or content patterns of a group (or group subset), which enables an efficient visual comparison among groups.

Integrated analysis. We introduce an integrated analysis module that is both data- and user-driven. We utilize graph clustering and indexing algorithms to generate an initial hierarchical structure for users to efficiently locate a group of interest from a large dataset. We then integrate a recently developed graph mining algorithm [20] to generate recommendations and help users query and refine a group with flexibly specified criteria.

We demonstrate the effectiveness of our system through case study and in-depth expert interview. The results suggest that our system is particularly useful in exploring rich multivariate graph data, and in tackling non-trivial criteria for forming a group. Our research sheds light on the design of systems for guiding users to explore the possibilities of "organizing things" in a complex setting through leveraging algorithmic solutions.

Throughout the paper, we use the following scenario to illustrate the problem context of interactive group mining and to motivate the system design. **Running example:** In a large-scale company, a manager often needs to build a task force – a team created for a specific project or to solve a particular problem (such teams may vary in size, e.g., ranging from tens to hundreds of members). An effective team requires members with the competencies or complementary skills to carry out the task, and more importantly, who can collaborate well with one another. For example, the manager may want to find members who have successful collaboration among themselves, or find one or more key people that have collaborated well with subsets of the team and can coordinate the communications across subgroups in the team. Hence, the essential criteria for building a team include both attribute (e.g., skill sets) and relational (e.g., collaboration relationship) aspects. Sometimes assembling a team involves more complex considerations such as team members' resource capacity or conflicts of interests. The manager would wish to find a proper set of members by incorporating these complex criteria.

RELATED WORK

We review research related to our work, from the aspects of *visualization* and *analysis*. We focus on (1) visualizing multivariate graphs, (2) visualizing sub-structures and groups, and (3) group mining techniques and applications concerning the use of multivariate networks.

Visualizing and Exploring Multivariate Graphs

Previous work on visualizing multivariate graphs focused primarily on how to represent both the topological structure of the graphs (networked data) and the multivariate attributes of nodes on the graphs. Most of this research leverage existing designs that had been originally proposed for displaying multivariate datasets. For example, to visualize the node or edge distributions in a social network, Bezerianos *et al.*'s GraphDice [2] used an interactive scatterplot matrix, and Shannon *et al.* [25] adopted parallel coordinates [17]. Wong *et al.* [31] used a pixel-based multivariate data visualization to represent node attributes. Xu *et al.*'s GraphSpace [32] is

a hybrid visualization that overlays the graph on top of a 3D density map generated using node attributes. In order to tease out the structure of multivariate graphs, visual techniques recently have started employing analytical operations based on node attributes. Lin *et al.* [22] and Cao *et al.* [6] introduced multi-relational graphs that can be used to represent different types of relations involving different node attributes, and Pretorius and Van Wijk [23] developed a technique that allows users to partition or aggregate the graph according to the attributes of nodes or edges.

As for exploring multivariate graphs, various interaction techniques have been proposed. Aggregation of nodes and links based on their attributes are the most common interaction designs, e.g., Pivot Graph [30] and Graph Trail [11]. GraphDice [2] switches between views of different node attributes by rotating a hypercube of attributes via animated transitions. DICON [5] supports splitting and merging functions to enable a flexible attribute summary of any subset of nodes or clusters. General exploring tools like search and filtering have been commonly used for exploring graphs or multivariate graphs [7, 12, 27, 29]. Although these techniques pointed out various possible ways for visualizing and exploring multivariate graphs, they are limited in offering functions to support group mining on multivariate graphs.

Visualizing Sub-structures or Groups in Networks

There has been work focusing on visualizations or exploration of sub-graphs, graph communities or clusters – which can be generally considered as groups. Traditional techniques for visualizing set data can be applied to visualize groups. For example, drawing a "convex hull" [14] of subsets of nodes on top of a graph is a commonly used method. Collions *et al.* [9] introduced BubbleSet that wraps the items belonging to the same set via a non-convex shape to avoid incorrect inclusion of set members. Although these are powerful representations for group membership in a graph, they do not capture richer aspects such as attribute differences of groups and hence cannot be applied directly to compare groups in a multivariate graph. Recent development of iconic representations for clusters makes it possible to compare and interpret clusters in terms of their within-group structures or features [5, 16]. Particularly, Henry *et al.* designed NodeTrix [15] in which graph clusters are shown as adjacency matrices to highlight the relational patterns and can be embedded in a node-link diagram to facilitate tracing links among clusters. Cao *et al.* introduced DICON for exploring feature patterns of groups in multivariate datasets [5]. Our work extends these iconic visualization designs to facilitate efficient characterization, comparison, and manipulation of groups in the process of interactive group mining.

Group Mining Techniques and Applications

A substantial body of graph mining research has focused on detecting groups or communities in graphs [13].Rather than based on pure topological structure, the problem of team assembly concerns extracting groups by considering both connections among and attributes of members [1, 10, 19]. It has been observed that to ensure a team's success, team members should possess the desired skills and have strong team cohesion. In other words, both relational (team members' social ties) and attribute (individuals' skills) structures need to be considered [10]. Such multi-objective requirements have been tackled recently by more sophisticated algorithms [1, 19]. However, the solutions are usually restricted into a ranking list of teams (or team members) based on fixed, predefined criteria, which prevents users' feedback to refine the mining results, or to dynamically incorporate more complex criteria such as the availability of team members.

Our proposed g-Miner goes beyond these solutions by designing an interactive group mining framework that allows users to flexibly and dynamically specify a rich set of criteria to build and refine groups iteratively. We leverage a recently developed algorithm proposed by Li *et al.* [20] which efficiently finds replacements for a given team based on both relational and attribute features. Our extension of this algorithm allows users to incrementally refine any chosen groups, with a new set of interactive visualizations specifically designed to guide users in the interactive group mining process.

SYSTEM REQUIREMENTS AND DESIGN

To better understand the needs for interactive group mining in multivariate graph data, we conducted a pilot study with expert users. Based on their feedback, we identified requirements that guide the design of the proposed system.

Pilot Study and Design Goals

Our pilot study is a multiple-session design process involving two expert users who have worked on team assembly research using social network methodology and machine learning approaches.

Session 1. Initial requirement: We discussed the challenges they encountered during the analysis. The experts pointed out that the most difficult task in their analysis is to *make sense of the results generated by different team assembly algorithms, especially when data involves team members' multivariate attributes and social relationships*. The initial requirement was to design a system that allows users to compare different possibilities of assembling a team with visualization for comparing members' attributes and connections.

Session 2. Prototyping: Based on the initial requirement, we develop the antecedent version of g-Miner.In this prototype system, we visualize a group using a multivariate graph representation where each team member is represented as a circular shaped voronoi diagram (similar to DICON [5]) with each cell representing a kind of expertise of the member, and members are connected by links as in standard node-link diagram representation.

Session 3. User study: We worked with the experts to conduct a preliminary user study. We recruited 20 people to participate in a team selection task using the prototype system. The system generated a set of team assembly results (through different algorithmic configurations such as "relation only," "skill only" and "proposed method") and asked the participants to compare and select the best team results (without knowing which result came from which method). The ages of the participants ranged from 22 to 35, all were college educated, and 2 of them were female. After the completion of the task, we asked users to provide free-form feedback on

the system interface. The preliminary study results showed that ,compared with the best alternative choices, our proposed best method achieved 27% and 24% net increase in average recall and precision, respectively. Users' informal feedback suggested our initial design allows users to compare teams by viewing the team members' expertise and connections simultaneously. However, several users commented that functionality is limited and only suitable for making straightforward comparisons between small teams. The feedback led to further discussion with the experts and helped to clarify the limitations and needs for team analysis tasks.

Through the multi-session pilot study, we identify the system requirements and design goals as follows. The first three requirements concern a system's capacity to support group mining tasks in general, followed by three requirements that further specify the goals of group visualization design.

R1 Scalability. Group mining and analysis is most valuable when users have to deal with a huge and heterogeneous dataset. A scalable system that allows users to explore groups from big data, big groups, or small groups within larger groups, is critical.

R2 Locating groups of interest. When dealing with a huge dataset, it is important for users to be able to navigate the entire dataset and quickly identify groups they may be interested in giving a closer inspection.

R3 Iterative group refinement. When an identified group is not satisfactory, users can specify or modify the criteria for making a better group. The system needs to incorporate such users' feedback to efficiently refine the group.

R4 Group abstraction. Groups need to be represented in a way that users can easily spot similarities or differences between groups, as well as to directly manipulate groups.

R5 Group characterization. Groups are formed by members based on their multivariate attributes (e.g., expertise) and connections (e.g., collaboration). The visual representation of a group should highlight different characteristics of a group in terms of these features. This can facilitate efficient group identification and comparison. For example, users can quickly recognize whether a group consists of members with similar or rather complementary expertise, or whether a group has densely connected members.

R6 Level of group comparison. Groups can be compared at different levels, from the attributes of their members, to member composition and connectivity within groups, to connections with other groups. The system should enable making different levels of comparison among groups.

Interactive Group Mining

We design an *interactive group mining* framework as the core of our new g-Miner system to meet the system requirements. As illustrated in Fig. 2, this framework incorporates two components that enable users to easily locate a group of interests from a large dataset (**R1,2**) and to iteratively refine the group according to user-specified complex criteria (**R3**).

Finding initial group. Three different methods are provided for users to locate a group of interest based on different levels of knowledge and expectations about the desirable group:

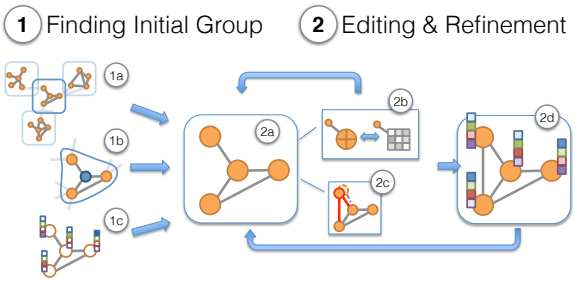

Figure 2. The process of interactive group mining.

Locating a group based on data hierarchy (1a): The system provides a *Hierarchy Explorer* that allows users to navigate the hierarchical structure of entire dataset. This data hierarchy is built based on the connections among individuals. Hence, selecting a densely-connected group from the data hierarchy is particularly efficient when users expect the desirable group to have dense connections among members.
Querying a member's ego-network (1b): If users consider more importantly the desirable group should include a particular member, they can simply query the member by name (Fig. 1(1)). When the system locates the member from the dataset, the ego-network centered on this member will also be shown. Here we show the so-called 1.5-degree egocentric network, i.e., a network consisting of all individuals having direct connections with this member and all connections among them.
Template matching (1c): If users are not concerned with whom should be included in the desirable group, but instead have specific criteria about the members' attributes, connections or overall group structure, users can use a *group editing tool* to flexibly specify complex criteria through building a *template group* and find a group from the dataset that best matches a template group.

Group editing and refinement. Once the initial group is found (2a), users can inspect the group by using the *Group Explorer* (2b). If the initial group found by the previous methods does not completely satisfy users' need, or users may decide to change the group criteria after seeing the actual data, users can refine the group (2c) by further editing the attributes or connections between members. The system then finds a list of candidate refined groups based on the modified criteria and users can compare these candidate groups by using the *Candidate Explorer*. These refinement steps can be continued iteratively until users find a group better meets their expectation (2d).

VISUALIZATION AND INTERACTION DESIGN

This section presents our novel visualization and interaction design, including (1) a multi-level group exploring interface, (2) the multi-structure group viewers, and (3) a group editing tool for users to flexibly specify the criteria of forming a group. We then provide a use case scenario to demonstrate the functionalities of these novel tools.

Multi-level Exploring Interface

As shown in Fig. 1, the user interface comprises three different exploring tools that allow users to explore and compare the group data at different levels:

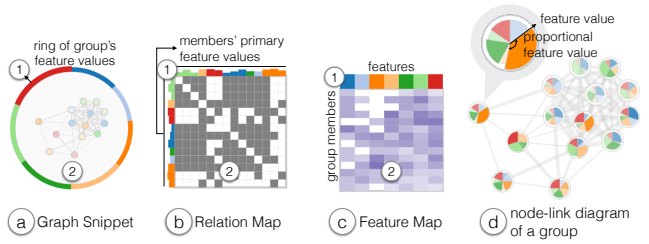

Figure 3. Visualization for characterizing a group: (a) Graph Snippet, (b) Relation Map, (c) Feature Map, and (d) Node-Link Diagram.

Group Explorer (Fig. 1(3)) represents a group of interest that can be selected from a hierarchical graph representation [12, 4, 26]. We use a fast greedy graph clustering algorithm [24] to hierarchically partition the entire network into groups, sub-groups and so on. Such a representation [12] offers an intuitive way to effectively make a big graph navigable (**R1**) and allows users to inspect the sub-structure of the group (**R6**). Alternatively, the focused group can be further presented by the multi-structure visualization tool (**R4,5**) detailed in the next subsection.

Hierarchy Explorer (Fig. 1(4)) visualizes the hierarchical clusters in an icicle tree view [18], providing an overview of the data hierarchy and supporting fast navigation (**R1**). In the tree view, leaves are individuals and the intermediate nodes represent clusters. Clicking any intermediate node makes the corresponding cluster a focused group to be visualized in the Group Explorer. Users can then narrow down the search and focus on replacing the undesirable part of the focused group (**R2,3**). This design follows the "overview first, zoom and filter, then details-on-demand" visualization guideline [28].

Candidate Explorer (Fig. 1(7)) allows users to compare the focused group with the list of candidate groups recommended by the system based on users' specified criteria. Users can switch between these candidates. Candidates are shown via multi-structure visualization as the focused group. The consistent representation of multiple groups enables detailed comparison at different levels of group structure in iterative group refinement (**R3,6**).

We proposed a multi-structure visualization design to represent the focused and candidate groups in both Group Explorer and Candidate Explorer.

Visualization Design

The multi-structure group visualization combines three different types of iconic views: Graph Snippet, Relation Map and Feature Map. The iconic views are designed to capture the characteristic properties of a group by encoding different information aspects, such as the graph topological structure, sub-structure, and attribute distribution, into a compact representation to facilitate group summarization and comparison (**R5**). We use a consistent visual encoding scheme across all three views: color hues encode different attributes (as in Fig. 1(5)), color opacity encodes values of attributes, and icon size encodes number of members of a group. At the same time, the iconic representation allows users to directly manipulate groups (**R4**): dragging to move a group, clicking to zoom-in the next level of the group, and selecting multiple sub-groups to aggregate their characteristic properties. We now describe the details of the three iconic representations.

Graph Snippet (Fig. 3(a), Fig. 1(a)) uses a node-link diagram that intuitively captures a group's topological structure. We generate a thumbnail of a node-link diagram (Fig. 3(a2)) and pack this thumbnail inside a ring with a circularly surrounding bar chart summarizing the attribute values of the group (Fig. 3(a1)). The ring sectors with different colors indicate different attributes. The arc length of each sector is proportional to the corresponding within-group attribute value. The height of each sector indicates the attribute value normalized across all groups in the dataset, which facilitate a comparison within and across different groups. Each node in the diagram is either a sub-group or an individual (i.e., a leaf in the data hierarchy). When the node represents an individual, it is visualized as a pie-chart-like icon (Fig. 3(d)). Similar to the ring encodings, each pie area encodes the corresponding attribute value relative to the member's other attribute values, and the height of each sector indicates the attribute value normalized across all individuals in the dataset.

Relation Map (Fig. 3(b), Fig. 1(b)) extends the design proposed in NodeTrix [15] to capture a group's relational patterns. It shows a matrix reflecting a group's connections among members. In this matrix, each row and column represent a member node in the graph and the opacity of each matrix entry encodes the connection strength between a pair of nodes in the group (Fig. 3(b2)). We visualize each node's primary attribute value via a small rectangle attached on the top and left of the corresponding row and column (Fig. 3(b1)). The rectangle is colored according to the primary attribute and sized by attribute value. To enhance visual patterns, we reorder the rows and columns of the matrix either by their primary attributes or by innate connection of the nodes inside the group (reordering methods can be chosen by users).

Feature Map (Fig. 3(c), Fig. 1(c)) employs a heatmap representation to capture the patterns of attribute values of each node in the group. In the heatmap, the header row shows attributes in different colors (Fig. 3(c1)). The remaining rows indicate nodes' attribute values encoded by color opacity (Fig. 3(c2)). The rows are reordered based on similarity of the feature vectors.

View Aggregation

The above view designs support flexible aggregation mechanism in which users can brush to select a set of (intermediate or leaf) nodes and groups to merge them together, forming a new group. A group can also be split into individual nodes by brushing it again. After aggregation, the views of the groups are updated based on the new set of nodes and links.

View Recommender

Each of the three views described before can be most suitable for capturing the characteristic properties of some groups but not others, depending on the structure of the group data. To help users choose a proper view, we further introduce an algorithm to automatically choose the most informative view for representing a group. Let a group G consist of n nodes with m attributes and their xy-coordinates indicating their locations on the two dimensional space. The information can

be formally described into three matrices: $\{\mathbf{A}_{n\times n}, \mathbf{F}_{n\times m}, \mathbf{X}_{n\times 2}\}$, where $\mathbf{A}$ is the adjacency matrix representing the connections, $\mathbf{F}$ is attribute matrix, and $\mathbf{X}$ is the 2D coordinate matrix. To make these matrices comparable, we compute the $n \times n$ similarity matrices $\mathbf{S}_F = \mathbf{F}\mathbf{F}^T$ and $\mathbf{S}_X = \mathbf{X}\mathbf{X}^T$ for estimating similarities of nodes corresponding to their attributes and coordinates respectively.

To determine the most informative view, we compute the informativeness score of an input matrix $\mathbf{M} \in \{\mathbf{A}, \mathbf{S}_F, \mathbf{S}_X\}$ based on information entropy. In brief, this algorithm computes an average entropy score of the rows in a row normalized matrix. The entropy of each row in the matrix suggests to what extent we can differentiate a node's neighbor based on the adjacency matrix or similarity matrices. Thus, a matrix with larger entropy is more "informative" (easy to be differentiated). Visually, this means the corresponding view has a more distinguished structure in terms of relations, attributes, or coordinates. When $\mathbf{A}$ is considered to be the most informative, we choose the Relation Map for a direct representation of $\mathbf{A}$. Similarly, when $\mathbf{S}_F$ is considered to be the most informative one, the Feature Map is selected. Otherwise, Graph Snippet is chosen to represent the group.

Group Editing

A set of group editing tools (Fig. 1(2)) are provided for users to flexibly specify criteria to form or refine a group: (1) *Member editing* allows adding or deleting member nodes in a group. (2) *Attribute editing* allows adjusting attribute values of each member in a group by clicking the sectors in the pie-chart-like icon. (3) *Relation editing* allows adding new edges between group members, removing existing edges from the group, or adjusting the weight of a selected edge via a slider. This function allows users to specify a particular link structure, e.g., to find a coordinate person with strong links with multiple sub-groups. View aggregation is supported in the editing mode.

Case Study with the Publication Dataset

We demonstrate the effectiveness of our visualization and interaction designs using a real-world paper publication dataset which contains multivariate graph information.

Dataset. This demonstration[1] is based on a paper publication dataset that comprises papers mostly in the Computer Science domain[2]. We transformed the data into a multivariate graph, in which nodes are researchers (authors) and edges are co-authorships with the weights representing the number of co-authoring papers. An author's expertise is represented by a seven-dimensional vector where each dimension represents the number of papers the author published in one of the seven research fields, including "data mining (DM)", "visualization (VIS)", "database (DB)", "nature language processing (NLP)", "artificial intelligence (AI)", "system", and "multimedia.". Both edge weights and expertise vectors are normalized for visualization purpose. The graph contains 78,997 nodes and 423,388 edges in total.

[1]The same dataset is used for conducting the interview with experts.
[2]http://arnetminer.org/download

Data Overview. The dataset contains four major research groups that are displayed as four big bubbles labeled by 1, 2, 3, and 4 (Fig. 4(A:a)). Each bubble contains several nodes in different colors, indicating sub-groups with different research interests. They are connected by links representing the collaborations across different groups or sub-groups.

Group Insights. Fig. 4(A:a) illustrates many high level differences of groups in terms of group size, primary research interests, and sub-structure. For example, group 4 is the largest group with three primary research interests including system (green), multimedia (brown), and database (orange). The sub-structure differences among groups can be captured in Fig. 4(A:b). For example, group 4 contains a central node that connects to all other nodes, reflected as a "cross" pattern in the matrix. The subgroups in group 3 are highly connected, shown as a clique in Figure 4(A:a) and as a dense matrix in Fig. 4(A:b). Both groups 1 and 2 contain multiple leading nodes (several crosses in the Relation Map). Fig. 4(A:c) displays group differences in terms of their research interests. For example, the members in group 3 have more diverse research interests as shown in Fig. 4(A:c), suggesting people in this group have comprehensive expertise.

As shown in Fig. 4(A:d), users can continuously zoom into their interested groups (e.g., group 4) guided by research topics (e.g., data mining and database), until reaching a cluster at the bottom of the hierarchy. From Fig. 1(3) users can clearly see that this is a star-shaped network centered on a person who has strong expertise in data mining and database. If the central person is unavailable, the team's coordination may be significantly affected. In this case, our system provides Candidate Explorer that allows users to query and find a most suitable candidate with similar skills and connections as the central person (Fig. 1(7)).

ANALYSIS: GRAPH MINING ALGORITHMS

We implement the group mining algorithm based on a fast Team Member Replacement algorithm ("TMR algorithm" for short) proposed by Li *et al.* [20]. The original TMR algorithm was designed to search the best replacement of a team member from multivariate graph data. The authors proposed the approximation method and demonstrated that this algorithm can efficiently generate high quality results in very large real-world datasets. This feature uniquely meets the scalability requirement (**R1**). This section first summarizes the key ideas in TMR algorithm, and describes how the algorithm is extended to support two core group mining tasks.

Team Member Replacement Algorithm

Considering a team in which a team member p is about to leave and the team needs a replacement, the goal of TMR algorithm is to find a similar person q to replace the current team member p. An ideal replacement q should not only have a similar skill set as p, but also have the proper connections with the rest team members so that the whole team can work together harmoniously.The algorithm was designed to solve the two-fold problem: (1) *skill matching*: the new member should bring a similar skill set as the current team member p; (2) *structural matching*: the new member should have a similar network structure as team member p in connecting

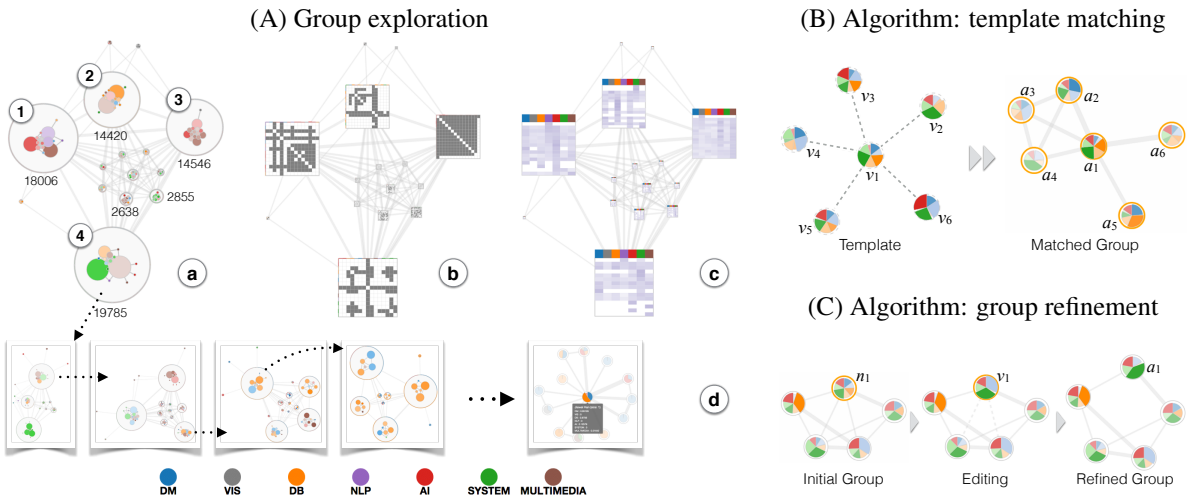

Figure 4. (A) An example of exploring groups in publication data using g-Miner. (B) An illustration of template matching. (C) An illustration of group refinement.

the rest of the team members. The objective can be formally described as:

$$q = \arg\max_{j \notin \mathcal{T}} \; Ker(G(\mathcal{T}), G(\mathcal{T}_{p \to j}))$$

where $\mathcal{T}$ is the set of nodes representing the members in the given team, $G(\mathcal{T})$ is the induced subgraph by the set $\mathcal{T}$ from a multivariate graph where each node is associated with a vector representing the values of the node attributes (e.g., skill set), and $G(\mathcal{T}_{p \to j})$ is the multivariate subgraph after the team member p is replaced by another individual j in data. The function $Ker(\cdot)$ is the graph kernel between these two labeled graphs. The intuition behind this optimization objective is to make the new team after the replacement as similar to the original one as possible. Here, a graph kernel function is used to compute the similarity between the two input graphs by considering both team member skills and relationships. A random walk graph kernel [3] is used in the algorithm implementation. The above model can be efficiently solved via an approximate algorithm and more details can be found in [20].

We extend the original TMR algorithm to support two core group mining tasks in g-Miner: (a) template-based group builder, and (b) group refinement recommender.

Template-based Group Builder

The template-based group builder allows users to build a group from scratch using a template graph. The template graph can be flexibly generated by using the graph editing tools provided in g-Miner. It is used to specify the desirable expertise features of group members (the attribute or content of nodes) and the desirable relational features within the group (the graph structure defined by the edges). Given the input template graph, the algorithm finds a subgraph that is most similar to the features defined in the template graph. This novel function enables users to explore a wide range of different possible group combinations from the data. It is particularly useful, for example, when a project leader seeks to create task forces for new tasks.

As shown in Fig. 4(B), the input of template-based group builder is a template graph consisting of virtual nodes (i.e., nodes not in the actual network) and edges. The goal is to find a subgraph in the data that best matches the given template. Note that if no such nodes exist in the data with the exact skill and relation configuration specified as in the given template, the output will be a subgraph that most resembles the given template.

To find the best matching subgraph, we iteratively find a match for each vertex at a time. This is to avoid dealing with the combinatorial problem – that is, to compute a similarity score for every possible subgraph in the network with the same size.

Group Refinement Recommender

Group refinement recommender is designed to refine an existing group by replacing some of the group members selected by users. For example, users may seek to find a person outside a team with better or similar connections and skills to replace one or more members who cannot be in the team for certain reasons (e.g., conflicts of interests or resource capacity).

As discussed in our pilot study, evaluating the results derived from an automatic approach for team refinement is challenging. More importantly, users may need to change their evaluation criteria flexibly and iteratively in the context of different applications (**R3**). Our group refinement recommender is designed to address these issues.

We use the TMR algorithm along with the interactive group mining interface to recommend a set of best replacement candidates. When users specifically add a new member with particular skills and connections, the TMR algorithm cannot be directly applied as it is designed to replace a current member with another one in the dataset, rather than "refine" the team with desirable member properties. We provide a solution similar to our template-based group builder: to find a candidate that best matches to a virtual group member with desired attributes and connections. This virtual group member can be generated by editing the member that is going

to be replaced in the existing group (Fig. 4(C)). The editing includes adding or removing links of the node and adjusting its attributes to the desired values.

EVALUATION VIA EXPERT INTERVIEW

This section presents our evaluation through an in-depth expert interview process.

We conducted interviews with three domain experts to fully understand the design and functionality of the proposed system. Two experts were first interviewed (interview study I & II), helping diagnose usability issues in interaction components and the overall mining pipeline. The identified usability issues were addressed before we conducted the interview with the third expert (interview study III), who provided feedback on the final design.

Procedure. We started each interview with a tutorial explaining the goals and features of g-Miner followed by a detailed demonstration of the system. The experts were then asked to use the system[3]. After they fully explored the system's functionality, we conducted a semi-structured interview covering questions on aspects of visual design, interactive group mining, overall usefulness, ease of use, and general pros and cons. Further, as the three expert users have different backgrounds and expertise, we asked them to elaborate on their opinions based on their expert domain knowledge. Each interview study lasted approximately 1.5 hours. Interviews were recorded and notes were taken.

Summary of interview study I & II. The first expert is a professor who also had worked with us for developing the prototype system in the pilot study. Familiarity with our system requirements allows him to offer detailed assessment of the new system. The second expert is a senior manager in an IT company. His team developed the first enterprise-wise expert recommendation system, SmallBlue [21].

Both experts were impressed by the system's capability, particularly on offering the interactive group mining capacity that allows mining large graph in real time, and on the amount and levels of information shown in the visualization. Expert #2 especially appreciated the idea of leveraging group mining algorithms to enable iterative group refinement, which he believed "a very powerful function that is missed in the SmallBlue system". Expert #1 noted the advantages of the multi-structure group visualization: "helps provide a more comprehensive picture of the sub-graphs from various aspects". Both experts described the iconic summary "can easily identify the differences [of groups]." Moreover, both experts commented that the Hierarchy Explorer and Group Explorer are the most efficient tools for fast data navigation.

Lessons learned from studies I & II. The two experts also raised several usability issues potentially undermining the interaction support in the group mining framework. After collecting their feedback and identifying the key usability issues, we incorporated solutions to address these issues into a final design as shown in Fig. 1. We summarize the key issues and our solutions as follows.

1. *Parallel views are not always useful.* The system we used in the first two interview studies inherited a similar look as the prototype system , which comprises Group Explorer and Candidate Explorer on two parallel aligned panels. Both experts believed the two views may not be used simultaneously. **Solution**: In our final design (Fig. 1), we reserved most of the visual space for Group Explorer and placed the Candidate Explorer into a floating window (Fig. 1(7)). This window will only show up when needed.The size of this floating window is the same as the size of Group Explorer. Users can drag this window on top of the Group Explorer and make a 1:1 visual comparison between the focused group and the recommended group in the Candidate Explorer.
2. *Missing hierarchical context while zooming-in.* Expert #2 pointed out that when he zoomed into the next level of a group, the context of the upper level is completely missing. **Solution**: We added an inset window inside the Group Explorer to display the upper-level graph of the focused group when applicable (Fig. 1(6)).
3. *Too many interaction functions to learn.* Expert #1 highlighted this learnability issue. **Solution**: We simplified the interactions in the entire group mining pipeline. The new interaction design simply requires standard mouse operations, such as dragging and clicking.

Summary of interview study III. The third expert is a research consultant working for an international IT company with more than 200,000 employees. A major part of her job is building teams in which the members can effectively propose IT solutions for resolving various customer problems. The customers come from diverse domains including Healthcare, Finance, etc., and the final solution usually covers the aspects of hardware, software, and services. She has worked in this position for more than five years and has abundant knowledge and experience in identifying experts from over 2000 employees with various background to form effective teams that can support her work. The dataset we used in the study has similar contexts as the data she is used to dealing with (both have individuals' expertise information and the collaborations among them). She was asked to build research teams using the final version of g-Miner and provide comments.

She immediately appreciated the design of g-Miner. After seeing our demonstration, she confirmed that the system has great novelty compared with any existing tools she knew: "existing tools I used can only recommend experts, but it is far from enough to build a temporary consulting team as the experts are usually unavailable [because of the conflict of the schedule, lack of interests, etc.]" She believed this system will be very useful for her to build teams and find replacement when candidates are unavailable. In the exploration process, she first chose to query a person, a data scientist whom she knew to start with. After briefly exploring the person's ego-network in the Group Explorer, she then removed some of the team members and replaced existing members with some new members iteratively via Group Editing and Refinement Recommender. She commented when using the tools: "it

[3]The system was developed using HTML5, JavaScript, D3.js (front-end) and Python CGI (back-end) and was deployed on an Amazon EC2 server. The expert users test the system by using Chrome browser with a 27-inch iMac.

runs very fast", "it is very nice that I can edit the attributes and connections of the team members to enter my requirements", and "it really finds a candidate for me [to replace another person in the team], and this guy is even better!"

Her response to the question, "which is the most problematic part of the system?", is very valuable. She felt building a team based on the Template Matching is a nice feature; however, the current implementation is not sufficiently effective because "it does not start with the people I'm familiar with" and "the system doesn't allow me to add multiple people that I know well into a template at the same time". She also pointed out three issues about the visualization design: (1) The outer parts of the Graph Snippet and Relation Map (expertise ring and bar chart, respectively) still require some effort of learning to understand. However, she agreed that these designs are extremely informative once the users learn about what is encoded in the visualization. (2) Comparing groups in the same window (e.g. Group Explorer) is sometimes difficult when group sizes are highly uneven. This leads to the readability issue: the iconic views of small groups may not be readable when a very large group exists in the same viewing space. (3) Too little textual information: Additional context information such as statistic of groups and topic keywords describing a person or a group's skills can also be very useful.

DISCUSSION

In the real world, team assembly requires tremendous human judgment and contextual knowledge that goes beyond the power of automatic learning and mining algorithms. Our g-Miner aims to facilitate the process through an interactive group mining system. The set of functionalities offered in g-Miner was mostly well accepted by the domain experts we interviewed. However, the current design still has limitations. We discussed these limitations and ways to improve them.

Bridging query and template-matching. As noted by expert #3, the current implementation of g-Miner does not support matching a template graph that comprises one or multiple default members. The current query function only allows users to find a single individual from the data and fetch his or her existing ego-network. This restricts users' options to either start with a (single) selected member (by querying), or specified structure (by template-matching), but not both. A more powerful query mechanism can be achieved by combining both query and template-matching such that when building a template, users can query and assign members to any nodes on the template.

Overcoming the readability of iconic representation. Our visualization design employs iconic representation to offer an abstract but characteristic view of a group in order to facilitate group manipulation and comparison. The iconic representation has a resolution limit as it captures group (topological or attribute) structure independent of the visualized icon size. This issue can be addressed in different ways: (1) When displaying groups with highly uneven sizes in the same view space, non-linear scaling (e.g., log-scale) should be used to enable more effective visual comparison among groups. Proper legends should be included to ensure the correct interpretation of icon size. (2) Interaction techniques such as fisheye and focus+context [8] can be used to enlarge a small icon with users' focus.

More details are on demand. Although g-Miner provides a large set of functionalities to make group mining more efficient, users usually need more detailed information to support their decision making after finding a group. This can be achieved by providing more context and information details or a statistical summary about the groups or group members (e.g., topic keywords, attribute values, group size, number of edges and more graph-based metrics) via additional views or tool-tips. By providing more detailed information when it is demanded in the context of exploration, the system can be potentially used as resource browser for discovering organizational knowledge.

Design implication for interacting with large multivariate graph data. Although this work focuses on addressing the technical challenge of team assembly, the graph-mining based interactive design solution can be widely applicable. On one hand, the interactive pipeline enables users to start with partial knowledge about a group and progressively form more complicated queries in the context of refining a group. On the other, the algorithms are able to take into account both users' new requirements and the group to be refined to generate a new solution and progressively refine the solution based on users' feedback. This idea of interactive group mining can also be applied to other scenarios that involve exploratory needs in heterogeneous data. For example, user-generated content, such as e-mails and photos, consists of rich information (e.g., topics and users' social networks) and users are often unable to determine how they wish to organize or explore the content beforehand. Our design concept can be potentially used in bridging users' partial knowledge and the capacity of advanced algorithms for tackling large heterogeneous datasets.

CONCLUSIONS AND FUTURE WORK

We have presented g-Miner, the first interactive group mining system that allows users to efficiently explore heterogeneous graph data and from which to progressively select and replace candidate members to form a group based on user-specified criteria. Our solution incorporates the design of interactive group mining pipeline, glyph-based group visualization and manipulation, and an integrated algorithmic component. Our system addresses the technical challenge of team assembly by connecting users with algorithmic solutions. In future work, we plan to explore new applications of the proposed framework. In addition, we plan to improve our current implementation by addressing the limitations discussed in the previous section, such as offering more flexible template-matching and incorporating proper interaction techniques to guide in-depth exploration.

ACKNOWLEDGMENTS

This material is partly supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under the Social Media in Strategic Communication program under Contract Number W911NF-12-C-0028, by the NSF under Grant No. IIS1017415, and by the Army Research Laboratory

under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U.S. Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

1. Balog, K., Azzopardi, L., and De Rijke, M. Formal models for expert finding in enterprise corpora. In *ACM SIGIR*, ACM (2006), 4350.
2. Bezerianos, A., Chevalier, F., Dragicevic, P., Elmqvist, N., and Fekete, J.-D. Graphdice: A system for exploring multivariate social networks. In *Computer Graphics Forum*, vol. 29, Wiley Online Library (2010), 863872.
3. Borgwardt, K. M., Schraudolph, N. N., and Vishwanathan, S. Fast computation of graph kernels. In *NIPS* (2006), 1449–1456.
4. Buchsbaum, A. L., and Westbrook, J. R. Maintaining hierarchical graph views. In *SODA*, Society for Industrial and Applied Mathematics (2000), 566–575.
5. Cao, N., Gotz, D., Sun, J., and Qu, H. DICON: Interactive visual analysis of multidimensional clusters. *IEEE TVCG 17*, 12 (Dec. 2011), 2581–2590.
6. Cao, N., Sun, J., Lin, Y.-R., Gotz, D., Liu, S., and Qu, H. FacetAtlas: Multifaceted visualization for rich text corpora. *IEEE TVCG 16*, 6 (2010), 1172–1181.
7. Chau, D. H., Kittur, A., Hong, J. I., and Faloutsos, C. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *ACM SIGCHI*, ACM (2011), 167176.
8. Cockburn, A., Karlson, A., and Bederson, B. B. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM CSUR 41*, 1 (2008), 2.
9. Collins, C., Penn, G., and Carpendale, S. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE TVCG 15*, 6 (2009), 1009–1016.
10. Cummings, J. N., and Kiesler, S. Who collaborates successfully?: prior experience reduces collaboration barriers in distributed interdisciplinary research. In *Proc. of 2008 ACM CSCW*, ACM (2008), 437446.
11. Dunne, C., Henry Riche, N., Lee, B., Metoyer, R., and Robertson, G. GraphTrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In *SIGCHI*, ACM (2012), 16631672.
12. Elmqvist, N., and Fekete, J.-D. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE TVCG 16*, 3 (2010), 439–454.
13. Fortunato, S. Community detection in graphs. *Physics Reports 486*, 3 (2010), 75–174.
14. Graham, R. L. An efficient algorith for determining the convex hull of a finite planar set. *Information processing letters 1*, 4 (1972), 132–133.
15. Henry, N., Fekete, J., and McGuffin, M. J. Nodetrix: a hybrid visualization of social networks. *IEEE TVCG 13*, 6 (2007), 1302–1309.
16. Henry, N., and Fekete, J.-D. Matlink: Enhanced matrix visualization for analyzing social networks. In *INTERACT*. Springer, 2007, 288–302.
17. Inselberg, A., and Dimsdale, B. Parallel coordinates for visualizing multi-dimensional geometry. In *International Conference on Computer graphics*, Springer-Verlag New York, Inc. (1987), 25–44.
18. Kruskal, J. B., and Landwehr, J. M. Icicle plots: Better displays for hierarchical clustering. *The American Statistician 37*, 2 (1983), 162–168.
19. Lappas, T., Liu, K., and Terzi, E. Finding a team of experts in social networks. In *ACM SIGKDD* (2009), 467476.
20. Li, L., Tong, H., Cao, N., Ehrlich, K., Lin, Y.-R., and Buchler, N. Replacing the irreplaceable: Fast algorithms for team member recommendation. *arXiv:1409.5512* (2014).
21. Lin, C.-Y., Ehrlich, K., Griffiths-Fisher, V., and Desforges, C. Smallblue: People mining for expertise search. *IEEE MultiMedia 15*, 1 (2008), 78–84.
22. Lin, Y.-R., Sun, J., Cao, N., and Liu, S. Contextour: Contextual contour visual analysis on dynamic multi-relational clustering. In *SDM* (2010).
23. Pretorius, A. J., and Van Wijk, J. J. Visual inspection of multivariate graphs. In *Computer Graphics Forum*, vol. 27, Wiley Online Library (2008), 967974.
24. Rosvall, M., and Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *PNAS 105*, 4 (2008), 1118–1123.
25. Shannon, R., Holland, T., and Quigley, A. Multivariate graph drawing using parallel coordinate visualisations. *University College Dublin, School of Computer Science and Informatics, Tech. Rep 6* (2008), 2008.
26. Shi, L., Cao, N., Liu, S., Qian, W., Tan, L., Wang, G., Sun, J., and Lin, C.-Y. Himap: Adaptive visualization of large-scale online social networks. In *IEEE PacificVis*, IEEE (2009), 41–48.
27. Shi, L., Liao, Q., Tong, H., Hu, Y., Zhao, Y., and Lin, C. Hierarchical focus+context heterogeneous network visualization. In *2014 IEEE Pacific Visualization Symposium (PacificVis)* (Mar. 2014), 89–96.
28. Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Symposium on Visual Languages*, IEEE (1996), 336–343.
29. van den Elzen, S., and van Wijk, J. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE TVCG 20*, 12 (2014), 2310–2319.
30. Wattenberg, M. Visual exploration of multivariate graphs. In *ACM SIGCHI*, ACM (2006), 811819.
31. Wong, P. C., Foote, H., Mackey, P., Chin, G., Huang, Z., and Thomas, J. A space-filling visualization technique for multivariate small-world graphs. *IEEE TVCG 18*, 5 (2012), 797–809.
32. Xu, W., Zhiquan, L., Kaili, Z., and Junying, G. Multivariate graphs representation model based on complex matrix. In *CSAE*, vol. 3, IEEE (2012), 50–54.