



REGULAR PAPER

Kun Fu · Tingyun Mao · Yang Wang · Daoyu Lin · Yuanben Zhang · Junjian Zhan · Xian Sun · Feng Li

TS-Extractor: large graph exploration via subgraph extraction based on topological and semantic information

Received: 1 July 2020 / Revised: 19 August 2020 / Accepted: 25 August 2020 / Published online: 22 September 2020
© The Visualization Society of Japan 2020

Abstract Exploring large graphs is difficult due to their large size and semantic information such as node attributes. Extracting only a subgraph relevant to the user-specified nodes (called focus nodes) is an effective strategy for exploring a large graph. However, existing approaches following this strategy mainly focus on graph topology and do not fully consider node attributes, resulting in the lack of clear semantics in the extracted subgraphs. In this paper, we propose a novel approach called TS-Extractor that can extract a relevant subgraph around the user-selected focus nodes to help the user explore the large graph from a local perspective. By combining the graph topology and the user-selected node attributes, TS-Extractor can extract and visualize a connected subgraph that contains as many nodes sharing the same/similar attribute values with the focus nodes as possible, thereby providing the user with clear semantics. Based on TS-Extractor, we develop a Web-based graph exploration system that allows users to interactively extract, analyze and expand subgraphs. Through two case studies and a user study, we demonstrate the usability and effectiveness of TS-Extractor.

Keywords Graph visualization · Visual exploration · Large graph exploration · Subgraph extraction

1 Introduction

Large graphs are widely used to describe relationships between entities, like collaborations between research institutions, friendships between people, communications between mobile devices, interactions between proteins, etc. Real-world large graphs are often complex. Aside from having thousands or more of nodes, they generally are accompanied with semantic information such as node attributes, which further increases the complexity of graph exploration. Visual exploration of large graphs provides an interactive

K. Fu · T. Mao · Y. Wang · D. Lin · Y. Zhang · J. Zhan · X. Sun · F. Li
Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China

K. Fu · T. Mao · Y. Wang · D. Lin · Y. Zhang · J. Zhan · X. Sun · F. Li
Key Laboratory of Network Information System Technology (NIST), Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

K. Fu · T. Mao · Y. Zhang · J. Zhan · X. Sun
University of Chinese Academy of Sciences, Beijing, China

K. Fu · T. Mao (✉) · Y. Zhang · J. Zhan · X. Sun
School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
E-mail: tingyunmao@gmail.com

visual means to help users explore associations between entities, discover interesting information and mine hidden patterns (Liu et al. 2014; Ghoniem et al. 2019; Zhao et al. 2019).

Various approaches for large graph exploration have been proposed. They can be separated into two main categories: global view-based and local view-based approaches (Pienta et al. 2015). Global view-based approaches display an overview of the whole graph and allow users to explore local details of interest by zooming or filtering (Herman et al. 2002). Although the overview can present global patterns to users such as outliers or clusters, it generally is visualized as a hard-to-read “hairball” which has excessive node overlaps and edge crossings. Faced with such an overview, users often do not know where to start exploration and which nodes and edges are relevant to their current tasks (Von Landesberger et al. 2011). Local view-based approaches show only relevant parts of a large graph given user-specified nodes of interest (called focus nodes), thereby avoiding the above problems (Chen et al. 2019). Such approaches need to extract a relevant subgraph (a local context) around the focus nodes so as to help users explore the large graph from a local perspective (Chau et al. 2011). Nevertheless, in many real-world scenarios, it is not easy to extract a satisfactory subgraph. Besides remaining connected, the extracted subgraph should contain as many nodes that are semantically relevant to the focus nodes as possible to provide clear semantics. As a concrete example, when exploring a citation graph in which nodes are papers and edges are citation relationships, a user would like to extract a connected subgraph that contains as many papers that share the same/similar keywords with the specified paper(s) as possible so that the user can investigate papers with similar topics and understand the relationships among them. To achieve the above goal, in addition to the graph topology, node attributes that reflect node semantics should be considered. When graph topology is considered alone, the extracted subgraph tends to contain nodes that are close to the focus nodes and have high degrees (Abello et al. 2014). However, those semantically relevant nodes, which have low degrees but share the same/similar attribute values with the focus nodes, cannot be captured. As a complement to the graph topology, node attributes can be considered as semantic constraints to help users capture semantically relevant nodes. However, existing approaches mainly consider graph topology and do not fully consider node attributes. As a result, the extracted subgraphs have no clear semantics due to the lack of semantically relevant nodes (Furnas 1986; Dupont 2006; Pienta et al. 2017). Moreover, considering both graph topology and node attributes (semantic information) for extracting subgraphs is also challenging, as the two distinct types of information need to be combined.

In this paper, we propose TS-Extractor, a novel approach that can extract a connected relevant subgraph around the user-selected focus nodes based on graph topology and node attributes to support large graph exploration. First, an augmented graph is constructed by using attribute values of the focus nodes in order to combine topology and semantic information. Then, Personalized PageRank is performed on the augmented graph to compute the relevance scores for nodes. Finally, the relevance scores are used to extract a relevant subgraph around the focus nodes. The extracted subgraph can provide users with clear semantics because it contains as many nodes that share the same/similar attribute values with the focus nodes as possible. We have developed an interactive graph exploration system to materialize TS-Extractor. Through TS-Extractor, we contribute:

- An approach to help users extract subgraphs relevant to their interests for local exploration of large graphs. It consists of a novel relevance computation, an effective diffusion method and an extended subgraph extraction algorithm.
- An interactive graph exploration system that implements our approach and provides interaction and visualization. It allows the user to select simultaneously multiple focus nodes, decide whether to consider node attributes and expand the extracted subgraph in a direction of interest. We demonstrate its usability and effectiveness through two case studies and a user study using real-world datasets.

The rest of this paper is structured as follows: We first discuss related work in Sect. 2. Next, we present the pipeline of our approach in Sect. 3. Then, we describe our visual designs and system implementation details in Sect. 4. We present two case studies and a user study to justify the usefulness of our system in Sect. 5. Finally, we conclude with a discussion in Sects. 6 and 7.

2 Related work

Our work focuses on how to facilitate visual exploration of large graphs. Visual exploration approaches for large graphs generally can be divided into two categories: global view-based and local view-based approaches (Herman et al. 2002; Von Landesberger et al. 2011; Pienta et al. 2015).

2.1 Global view-based approaches

Global view-based approaches visualize the entire graph to present users with an overview of the global graph structure. However, in addition to increasing the computational cost of graph visualization, the escalating graph size induces visual clutter, precluding users' exploration. At the data level, some methods such as sampling (Leskovec and Faloutsos 2006), filtering (Jia et al. 2008; Zhan et al. 2019) and edge cutting (Edge et al. 2018) can eliminate unimportant nodes or edges in a large graph to provide a more compact overview. At the visualization level, aggregating nodes or edges can reduce visual clutter to improve the readability of graph visualizations. Topology-based aggregation methods (Auber et al. 2003; Abello et al. 2006; Dunne and Shneiderman 2013; Li et al. 2017) use topological information to detect substructures (e.g., motifs) or clusters in a large graph and then aggregate them into abstract nodes to construct a high-level structure summary. Semantic-based aggregation methods (Shneiderman and Aris 2006; Pretorius and Wijk 2008; Stef et al. 2014) aggregate nodes with the same attribute values into abstract nodes to provide a coarse-grained abstract summary. For example, PivotGraph (Wattenberg 2006) helps users understand the relationship between node attributes and connections by rolling up a graph with discrete categorical node attributes (e.g., gender and location) into an abstraction graph. Topology + semantic-based aggregation methods (Shen et al. 2006) have been proposed. For instance, OnionGraph (Shi et al. 2014) creates five hierarchies by semantic and topological aggregation and allows users to drill down to lower hierarchies. Moreover, edge-based aggregation methods like edge bundling (Holten 2006; Hong et al. 2013) help users understand the structure of connections in graph layouts by binding the edges that are close together. At the interaction level, effective interaction techniques such as focus + context (Furnas 1986), Link Sliding and Bring & Go (Moscovich et al. 2009) can help users explore regions of interest in an overview. Although global view-based approaches help users understand global patterns such as clusters and outliers, they cannot tell users which nodes or edges are relevant to the users' current interests.

2.2 Local view-based approaches

Instead of visualizing the entire large graph, local view-based approaches present only relevant parts of the graph to the user given the user-specified nodes of interest.

The easiest method is to extract and visualize all or part of the neighbors of a specified node (Heer and Boyd 2005). However, it is difficult for the user to identify which neighbors are most relevant to the user's current interest from the visualization. FACETS (Pienta et al. 2017) can display the most interesting relevant neighbors of the user-specified node. It finds these nodes by their interest and surprise scores computed based on distributions of topological features (e.g., degree) or node attributes (e.g., number of papers for author nodes).

Node recommendation-based methods recommend relevant nodes based on topological information when the user specifies nodes of interest. Crnovrsanin et al. (2011) developed a visual recommendation system that can recommend nodes relevant to the selected node by using collaborative filtering and relevance filtering. However, the subgraph consisting of the recommended nodes may be disconnected because some recommended nodes may be far from the selected node. It is difficult for users to understand the associations between nodes in a disconnected subgraph. Apolo (Chau et al. 2011) uses the Belief Propagation algorithm to infer which nodes are relevant to the selected nodes. The inferred nodes and the selected nodes form a connected subgraph to support large graph sensemaking. Similarly, Refinery (Kairam et al. 2015) returns associated content using the random walk algorithm once the user selects known nodes.

Degree-of-interest (DOI)-based methods use their DOI functions to extract subgraphs that match users' interests. Computing DOI can be based on graph topology and node attributes. The idea of DOI was first introduced by Furnas (1986). However, Furnas's DOI function cannot be applied to general graphs. Frank and Adam (2009) extended Furnas's DOI function to graphs by adding a semantic component that is used to infer user interest from keyword search. Although their DOI function can help a user capture the context of interest, it only allows the user to select a single focus node in graph exploration. Recently, Laumond et al. (2017) extended Frank and Adam's DOI function to multilayer graphs. Their DOI function allows the user to continuously select multiple focus nodes until a satisfactory subgraph is extracted. Unfortunately, it does not consider the constraints of node attributes, resulting in the extracted subgraphs tending to capture the neighbors of focus nodes.

Yet, existing approaches do not fully consider the importance of node attributes for extracting relevant subgraphs. Our approach is an addition to existing local view-based approaches. It combines topology and

node attributes to extract a relevant subgraph that not only matches the user's interest, but also contains nodes sharing the same/similar attribute values with the user-specified nodes.

3 TS-Extractor

We first state the problems that TS-Extract needs to consider and define two key concepts for TS-Extract. Then, we introduce TS-Extract's pipeline through an example.

3.1 Problem statement

The input is a connected graph $G = (V, E, A)$ where V is a node set, E is an edge set and $A = \{attr_1, \dots, attr_m\}$ is the set of m attributes associated with nodes in V . An attribute $attr_j \in A$ of node $v_i \in V$ (e.g., the "keywords" attribute of a paper in a citation graph) has one or more attribute values. A large graph usually has multiple node attributes that may be numeric (e.g., age or height) or non-numeric (e.g., gender or grade). Our approach can support non-numeric attributes and numeric attributes simultaneously.

Given a set of user-selected focus nodes $Y \subset V$ called a focus set, our goal is to extract a connected subgraph that captures as many nodes that share the same/similar attribute values with the focus nodes as possible. With a semantic augmentation, the extracted subgraph matches the user's interest more closely. To achieve the goal, TS-Extractor needs to consider the attributes of the focus nodes. Moreover, the focus nodes may be selected from the search results obtained by keyword search. The search keywords are essentially certain attribute values of the focus nodes. Moreover, the search keywords as user input can reflect the user's interest or intention. Thus, TS-Extractor also takes into account them. However, TS-Extractor does not blindly consider all the attributes of each focus node or any search keywords. To make it flexible, we define two key concepts:

- *Attribute of interest (AOI)*. If all attributes of each focus node are considered, TS-Extractor's computation becomes rather complicated. In fact, the user may only be interested in certain attributes of each focus node. Thus, TS-Extractor allows the user to select the attributes he or she is interested in for each focus node, which makes the extracted subgraphs more interpretable. A selected attribute is defined as an attribute of interest.
- *Keyword of interest (KOI)*. The search keywords can reflect the user's current interest. However, some keywords such as node names or node IDs are useless for capturing the desired nodes. A keyword that can match multiple nodes is defined as a keyword of interest. TS-Extractor only considers keywords of interest.

3.2 TS-Extractor's pipeline

As shown in Fig. 1, we use an example to explain TS-Extractor's pipeline for extracting a relevant subgraph. The input graph $G_{in} = (V, E, A)$ has the set of node attributes $A = \{attr_1, attr_2, attr_3\}$. The pipeline consists

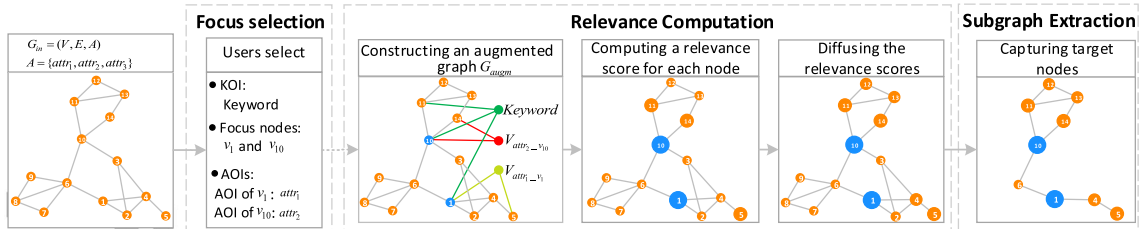


Fig. 1 TS-Extractor's pipeline illustrated by an example. Given the input graph G_{in} with node attributes $\{attr_1, attr_2, attr_3\}$, the nodes v_1 and v_{10} searched out by using the KOI *Keyword* (a value of $attr_3$) are selected as focus nodes (in blue). Meanwhile, $attr_1$ of v_1 is selected as the AOI of v_1 and $attr_2$ of v_{10} is selected as the AOI of v_{10} . The values of the AOIs of the focus nodes are denoted as $V_{attr_1_v_1}$ and $V_{attr_2_v_{10}}$, respectively. Next, *Keyword*, $V_{attr_1_v_1}$ and $V_{attr_2_v_{10}}$ are treated as semantic nodes and connected to their matching nodes (i.e., target nodes) to construct an augmented graph G_{augm} . Then, Personalized PageRank is executed on G_{augm} to compute the relevance score of each node in G_{in} (encoded by node size). To ensure that target nodes can be captured, the computed relevance scores are diffused. Finally, the target nodes around the focus nodes are captured by a greedy local search algorithm

of three major phases: focus selection, relevance computation and subgraph extraction. Based on the example in Fig. 1, we introduce these three phases below.

3.2.1 Phase 1: Focus selection

As the starting point of TS-Extractor, focus nodes can be searched out by keyword search or identified by graph centralities (PageRank, degree, betweenness, etc.). We use the KOI *Keyword* (assume it is a value of $attr_3$) to search for nodes of interest and then obtain results including nodes v_1 , v_{10} and v_{11} . The nodes v_1 and v_{10} are selected as focus nodes. Meanwhile, the attribute $attr_1$ is selected as the AOI of v_1 and the attribute $attr_2$ is selected as the AOI of v_{10} . For simplicity, we only select an AOI for each focus node. In fact, our approach allows users to select any number of AOIs for each focus node according to their interest. The values of the AOIs of the two focus nodes are denoted as $V_{attr_1-v_1}$ and $V_{attr_2-v_{10}}$, respectively.

3.2.2 Phase 2: Relevance computation

Both KOIs and the values of AOIs belong to semantic information associated with the focus nodes. In the relevance computation (Fig. 2), they are all treated as auxiliary nodes called semantic nodes. In this example, three semantic nodes (i.e., *Keyword*, $V_{attr_1-v_1}$ and $V_{attr_2-v_{10}}$) are added to G_{in} (Fig. 2a). Each one then is connected to its matching nodes in G_{in} (Fig. 2a). For instance, $V_{attr_1-v_1}$ is connected to v_1 and v_5 . If a semantic node is a non-numeric attribute value, we use similarity matching to find its matching nodes in G_{in} . However, if it is a numeric attribute value, we use a conditional matching to find its matching nodes such as nodes whose attribute values are no less than it. In this way, an augmented graph G_{augm} (Fig. 2a) is constructed. The newly added edges are called matching edges. If G_{in} is a weighted graph, we need to assign a weight to each matching edge to make G_{augm} a weighted graph. To this end, we treat all the matching edges equally and assign the same weight to each one. However, the weights of matching edges and the edge weights of G_{in} have different meanings. Therefore, we need to normalize the edge weights of G_{in} prior to assigning the same weight to all the matching edges. Common normalization methods such as min-max normalization or mean normalization can be used. However, we need to ensure all the normalized edge weights are nonzero. Otherwise, the corresponding edges will not exist. Based on the above considerations, we only divide each edge weight by the maximal edge weight. The normalization formula of edge weights in G_{in} is expressed as:

$$edge_weight_{norm} = \frac{edge_weight}{edge_weight_{max}}. \quad (1)$$

Next, we assign the value of 1 to each matching edge as its weight according to our experience. If G_{in} is an unweighted graph, it can be treated as a weighted graph whose edge weights are all 1. Then, we execute Personalized PageRank (PPR) (Haveliwala 2003) on G_{augm} to compute the relevance score of each node with respect to the starting nodes. Note that the starting nodes consist of the focus nodes and the semantic nodes (i.e., v_1 , v_{10} , *Keyword*, $V_{attr_1-v_1}$ and $V_{attr_2-v_{10}}$). The recursive equation of PPR is as follows:

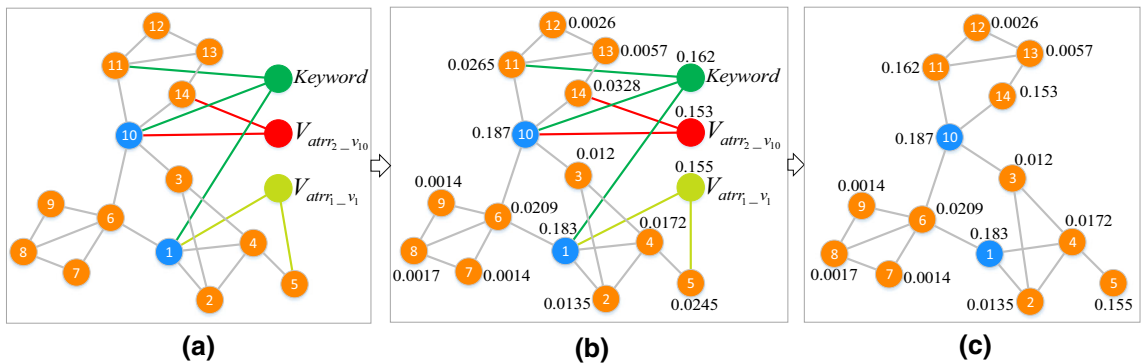


Fig. 2 The process of computing the relevance score for each node in G_{in} . Based on G_{augm} (a), (b) TS-Extractor uses Personalized PageRank to compute the relevance score of each node w.r.t. the focus nodes and the semantic nodes. c After the relevance score of each semantic node is assigned to its matching nodes (except the focus nodes), the relevance score of each node in G_{in} is obtained

$$\mathbf{p}^{t+1} = (1 - r)\widetilde{\mathbf{W}}\mathbf{p}^t + r\mathbf{p}^0, \quad (2)$$

where $\widetilde{\mathbf{W}}$ is the column-normalized weighted matrix of G_{augm} . In the restart vector \mathbf{p}^0 , the elements corresponding to the starting nodes are $1/m$ (m is the number of the starting nodes), while the other elements are 0. The default value for the restart probability r is 0.7. The steady-state probability vector \mathbf{p}^∞ is reached by performing the iteration until $\|\mathbf{p}^{t+1} - \mathbf{p}^t\|$ is less than 10^{-6} . The element $\mathbf{p}^\infty[v_i]$ represents the relevance score of node v_i with respect to the starting nodes.

The computed relevance score of each node in G_{augm} is shown in Fig. 2b. However, our final goal is to compute the relevance score for each node in G_{in} . The matching nodes of the semantic nodes are the target nodes that we want to capture. To ensure that the matching nodes (target nodes) are captured, we need to increase their relevance scores. Specifically, the final relevance score of a matching node v_i depends on the maximum of its current relevance score and the maximal relevance score of all semantic nodes connected to it:

$$\mathbf{p}_{final}^\infty[v_i] = \text{Max}(\mathbf{p}^\infty[v_i], \text{Max}(s \in S(v_i) : \mathbf{p}^\infty[s])), \quad (3)$$

where s is a semantic node, $S(v_i)$ represents all semantic nodes connected to v_i and $\mathbf{p}_{final}^\infty$ is the final relevance score vector. Each element in $\mathbf{p}_{final}^\infty$ is the relevance score of its corresponding node. As a supplement, we pre-compute the PageRank score (Brin and Page 1998) for each node of G_{in} to measure its importance. Thus, the relevance score for node v_i in G_{in} can consist of two components:

$$\text{Score}(v_i) = \alpha \cdot \mathbf{p}_{final}^\infty[v_i] + (1 - \alpha) \cdot \text{PageRank}(v_i), \quad (4)$$

where α can be used to adjust the weight of each component. PageRank is only computed once. $\mathbf{p}_{final}^\infty$ is recomputed whenever the focus set Y changes. PageRank score can help users capture important nodes around the focus nodes. However, in this example, we do not consider the PageRank score of each node (i.e., $\alpha = 1$). The relevance score of each node in G_{in} is shown in Fig. 2c.

Based on the computed relevance scores, we can utilize a greedy local search algorithm to find a relevant subgraph. However, there may be many nodes that have high scores but are surrounded by nodes with low scores, such as v_5 (Fig. 2c). These nodes are generally target nodes but may not be reached by the greedy local search algorithm. We address this problem by diffusing the relevance score of each node to its neighbors. Concretely, the final relevance score of node v_i is the maximum of its current relevance score and the maximal score diffused from its neighbors. Nevertheless, if v_i is a neighbor of a focus node, its final relevance score may only depend on the score diffused from the focus node because the relevance score of the focus node is usually much higher than that of each other neighbor of v_i (e.g., v_4 in Fig. 2c). Thus, if all neighbors of a focus node accept the scores diffused from the focus node, their final relevance scores may be the same, resulting in the greedy local search algorithm only reaching the neighbors of the focus node rather than target nodes. To avoid this problem, all neighbors of a focus node do not accept the scores diffused from the focus node. Besides, to control the direction and amount of diffusion, we define an edge importance function $EI(e, x, y)$ that can measure the importance of the edge e between node x and node y . The final relevance score for v_i is:

$$\begin{aligned} \text{Score}_{final}(v_i) = \text{Max}(\text{Score}(v_i), \delta \cdot \text{Max}(EI(e, v_i, n) \cdot \text{Score}(n))) \\ n \in N(v_i) \wedge n \notin Y, \end{aligned} \quad (5)$$

where $N(v_i)$ represents a set of neighbors of v_i , Y is the focus set and the diffusion factor δ ($0 \leq \delta \leq 1$) controls the amount of diffusion. We can increase the value of δ to enhance diffusion. For weighted graphs, $EI(e, v_i, n)$ is the weight of edge e , which can help us capture nodes connected by edges with high weights. For unweighted graphs, $EI(e, v_i, n)$ is equal to 1. Through diffusing scores (Fig. 3), the target nodes v_5 , v_{11} and v_{14} are captured.

3.2.3 Phase 3: Subgraph extraction

We can extract a relevant subgraph around the focus nodes according to the final relevance scores of nodes (see Algorithm 1). Our extraction algorithm is the extension of a greedy local search algorithm (Frank and Adam 2009). There may be multiple shortest paths between two focus nodes. The sum of the relevance scores for nodes in a shortest path can measure the interestingness of the shortest path. We define the

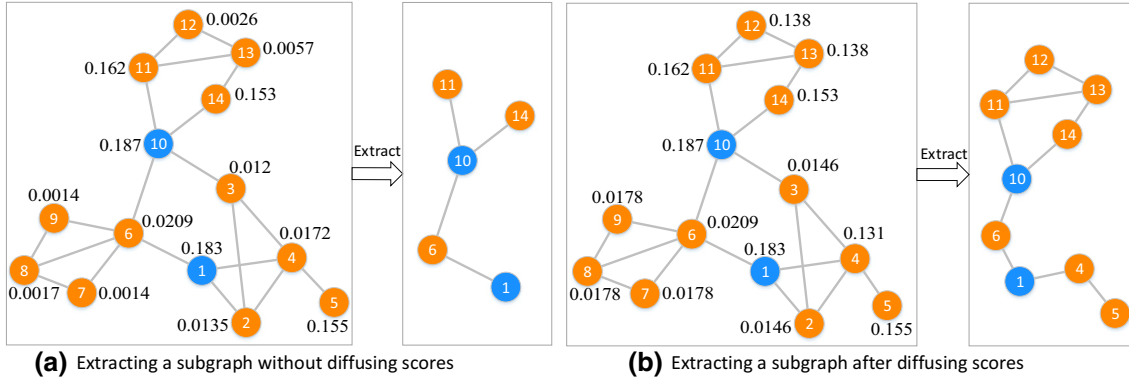


Fig. 3 An example illustrating the effectiveness of diffusion. By diffusing the scores, the target node v_5 also is captured

shortest path with the highest sum of relevance scores as a path of interest. Any two focus nodes can be connected by the path of interest between them to guarantee the connectedness of the extracted subgraph.

Algorithm 1: Extracting a relevant subgraph

Input: G : a graph with relevance scores; Y : a focus set; $ctrl_size_subgraph$: a constant for controlling the size of the subgraph
Output: g : a connected relevant subgraph

```

1 if  $G$  is a directed graph then
2   Convert  $G$  to an undirected graph;
3  $S_g = \{\}$ ; // Store all nodes of  $g$ ;
4 for each focus node  $y_i$  in  $Y$  do
5    $S = \{y_i\}$ ,  $C = \{\text{all neighbors of } y_i\}$ ;
6   while the size of  $S < ctrl\_size\_subgraph$  and  $C \neq \{\}$  do
7     Remove the node  $v$  with the highest relevance score from  $C$  and add  $v$  into  $S$ ;
8     Add all neighbors of  $v$  into  $C$ ;
9   Add all nodes in  $S$  into  $S_g$ ;
10  $Count = 0$ ;
11 for each focus node  $y_i$  in  $Y$  do
12   for each focus node  $y_j$  in  $Y$  do
13      $Count + 1$ ;
14     if  $y_i \neq y_j$  and  $Count \leq C_{|Y|}^2$  then
15       Find the path of interest between  $y_i$  and  $y_j$  and then add all its nodes into  $S_g$ ;
16 Extract the corresponding subgraph  $g$  from  $G$  according to  $S_g$ ; return  $g$ ;

```

As shown in Fig. 4, through comparing three different cases, we demonstrate that our approach can capture the corresponding matching nodes (i.e., target nodes) when considering semantic information associated with focus nodes (e.g., search keywords or node attributes). In these cases, we set $ctrl_size_subgraph = 5$, $\alpha = 1$ and $\delta = 0.85$.

4 The TS-Extractor system

We have developed a Web-based graph exploration system to implement the TS-Extractor approach. As shown in Fig. 5, the user interface of our system consists of five main views: (1) The **focus selection view** is used to help users select focus nodes. The selected focus nodes are shown in (2) the **focus set view**. (3) The **control panel** allows users to adjust parameters to extract a satisfactory relevant subgraph. (4) The **subgraph view** displays the extracted subgraph and allows users to explore and expand the subgraph interactively. (5) The **historical view** allows users to review exploration history.

Focus selection view selecting focus nodes is the starting point for graph exploration. To guide users to select nodes of interest, the focus selection view (Fig. 5a) provides two entrances:

- *Ranking List* When a graph dataset is loaded, the PageRank score for each node is calculated. The top 100 nodes ranked by PageRank scores are displayed in the ranking list (a1 in Fig. 5a). When a user is

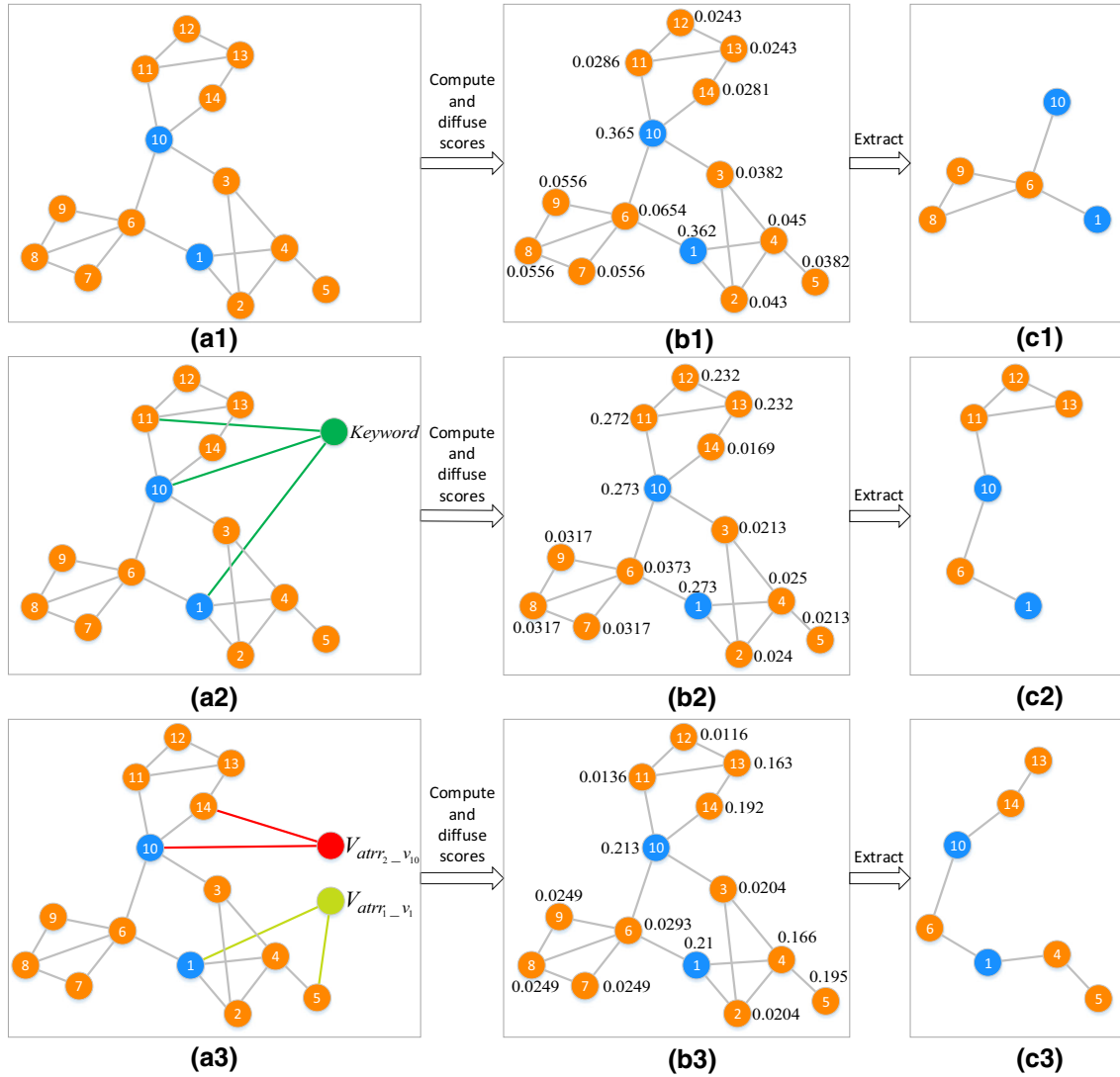


Fig. 4 Comparison of the three different cases: ignoring any semantic information associated with the focus nodes (a1), considering the KOI Keyword (a2) and considering the AOIs of the focus nodes (a3). When considering semantic information associated with the focus nodes (a2, a3), TS-Extractor can capture the target nodes (c2, c3)

unfamiliar with the graph dataset or does not have a clear exploration intent, the user can select focus nodes from this list. In addition to the default ranking metric PageRank, the system provides other options: degree, betweenness and closeness. Users can switch between these metrics via a drop-down menu.

- *Search box* When a user has clear objectives to explore, the user can enter KOIs in the search box (a2 in Fig. 5a) to search for related nodes. The search results are displayed in a search result list. However, if there are too many search results, it is difficult for the user to find nodes of interest from the search result list. The user can click the ☞ icon at the upper right of the list to unfold the search result view where the subgraph corresponding to the search results is displayed as a node-link diagram (Fig. 6). The size of each node in the subgraph represents its importance.

Users can select nodes of interest from the ranking list, the search result list or the search result view as focus nodes and then add the selected focus nodes to the focus set. The example in Fig. 6 illustrates how to add a focus node to the focus set.

Focus set view This view (Fig. 5b) shows all the focus nodes in the focus set. The user can hover over each focus node to inspect its details such as KOIs and AOIs. In addition, the user can delete focus nodes

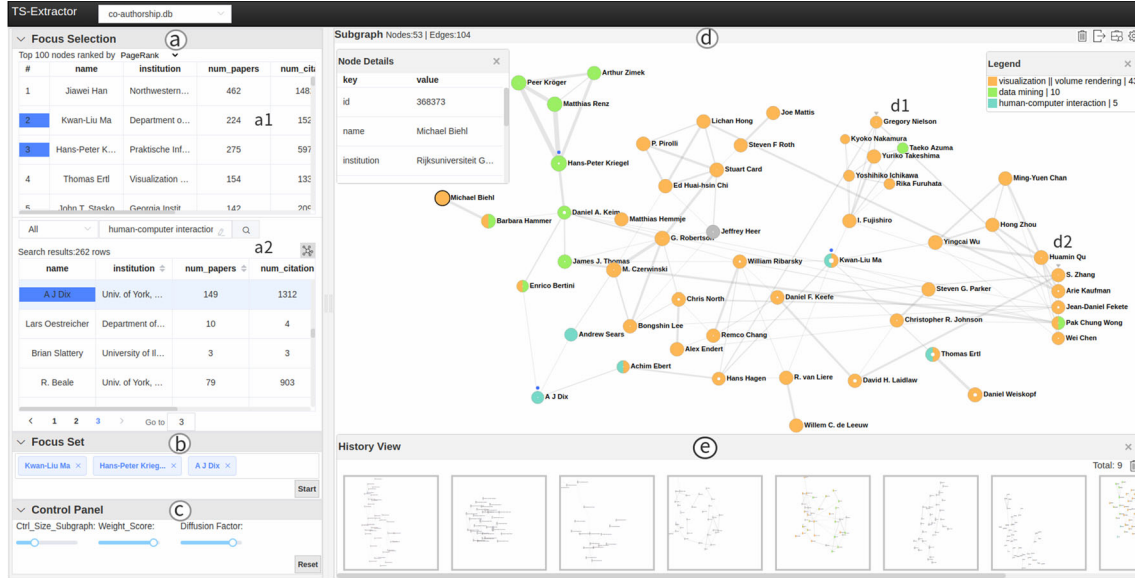


Fig. 5 TS-Extractor’s user interface showing the user exploring the co-authorship graph starting with three focus nodes (each with a blue dot at its top). **a** The focus selection view offers two focus selection entrances: **a1** a ranking list for showing the top 100 important nodes and **a2** a search box for keyword search. **b** The focus set view shows the user-selected focus nodes. **c** The control panel allows the user to adjust parameters. **d** The subgraph view displays the extracted subgraph. The size of a node is proportional to its relevance score. The size of the white dot at the center of a node encodes the number of the node’s current remaining neighbors. The current subgraph can be expanded by adding a node’s top N remaining neighbors ranked based on their attributes (e.g., paper count or citation count). Newly added neighbors can be placed by circular layout (**d1**) or vertical layout (**d2**). **e** The history view allows the user to review exploration history

from the focus set. After determining all the focus nodes, the user can click the “Start” button to extract a relevant subgraph.

Subgraph view The view (Fig. 5d) is the main workspace for graph exploration. The extracted subgraph is visualized as a node-link diagram. The size of each node is proportional to its relevance score. The user can click on each node or edge to inspect its details. We allow the user to select nodes of interest from the current subgraph as new focus nodes to re-extract a satisfactory subgraph. The view offers multiple interactions such as panning & zooming, dragging & dropping nodes and highlighting a node and its immediate context. To obtain new contextual information, the user can expand a node by adding its top N remaining neighbors ranked by their relevance scores or attribute values (e.g., paper count, citation count or h-index). We provide two methods to lay out the newly added neighbors. Circular layout (Fig. 5d1)) places each neighbor clockwise based on the size of the neighbor’s relevance score or attribute value. The neighbor with a small inverted triangle on the top is first placed. Vertical layout (Fig. 5d2) arranges neighbors vertically from top to bottom in the same strategy. When the user expands a node, a dialog box pops up allowing the user to set N and select a layout and an attribute used to rank its remaining neighbors. The number of current remaining neighbors per node is encoded by the size of a white dot at its center. The user can hover over a node to check the number of its current remaining neighbors. After a node is expanded, the number of remaining neighbors per node is updated. Analyzing nodes with the same or similar attribute values can help users mine hidden information. To find nodes sharing the same/similar attribute values with a specified node, we designed a matching panel (Fig. 7). The user can add one or more nodes from the subgraph to the matching panel and then assign each a matching condition consisting of its attribute value(s). The nodes matching the specified conditions are highlighted with halos in the corresponding colors. An example is shown in Fig. 7.

History view This view (Fig. 5e) can help the user review or restore exploration history at any time. When the user hovers over a thumbnail, the thumbnail is zoomed in and simultaneously the corresponding information (e.g., time, focus nodes and parameters) is shown. If a thumbnail is clicked, the corresponding subgraph is displayed in the subgraph view.

Control panel The control panel (Fig. 5c) can help users adjust the size and quality of a subgraph. As mentioned earlier, TS-Extractor involves several parameters. However, we only expose three parameters:

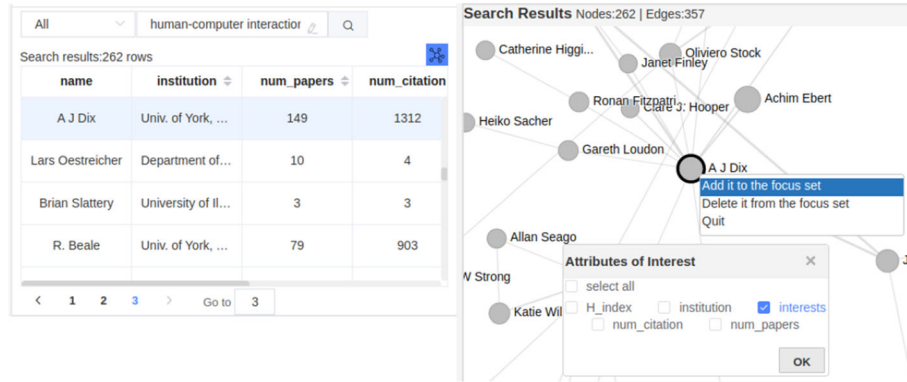


Fig. 6 An example of selecting a focus node from the search result view. The user first enters the keyword “human-computer interaction” to search for related authors (left). Due to a large number of search results, the user unfolds the search result view (right). To add the node *A J Dix* to the focus set, the user right clicks on it and selects the corresponding item from the pop-up menu. Next, the user decides whether to select AOIs for this focus node from the pop-up “Attributes of Interest” dialog. After the user clicks the “OK” button, this focus node is added to the focus set. Following the above steps, the user can add multiple focus nodes one by one

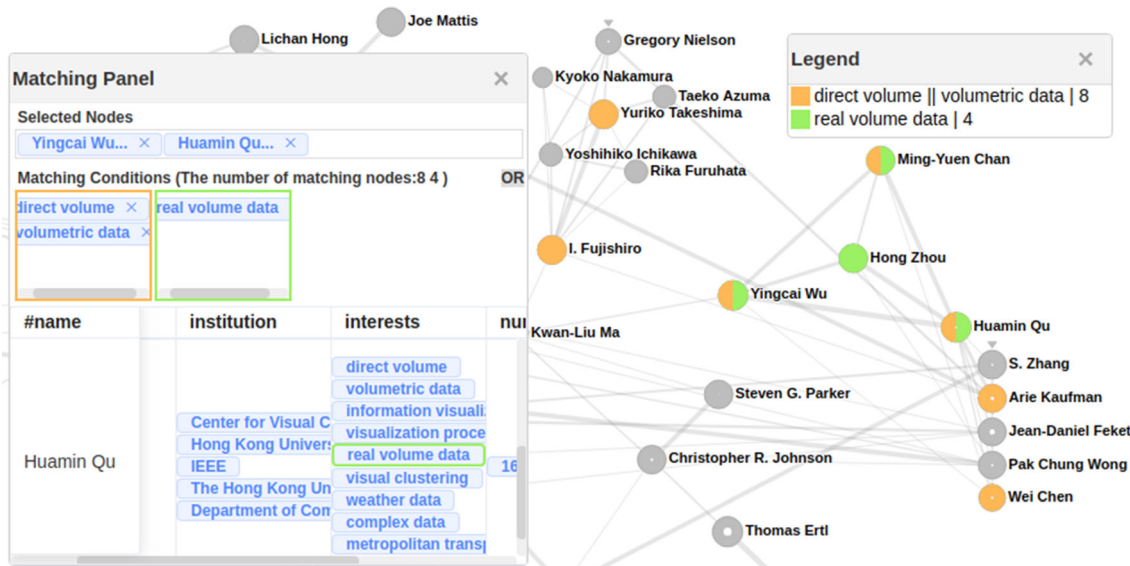


Fig. 7 An example using the matching panel. The authors *Yincai Wu* and *Huamin Qu* are added to the matching panel. The research interest “direct volume” and “volumetric data” of *Yincai Wu* are selected as the first matching condition (connecting each item with or logic). The research interest “real volume data” of *Huamin Qu* is selected as the second matching condition. Nodes that match these two conditions are immediately highlighted with the corresponding color (see the legend). The three authors *Ming-Yuen Chan*, *Yincai Wu* and *Huamin Qu* match these two conditions simultaneously

$ctrl_size_subgraph$ (see Algorithm 1), α and the diffusion factor δ . Unlike α and δ , adjusting $ctrl_size_subgraph$ will not result in recomputing the relevance score of each node. The user can adjust $ctrl_size_subgraph$ to change the size of the extracted subgraph in real time. Increasing α gives more preference to the relevance score. By adjusting δ , users can control the amount of diffusion. By default, the parameter settings are $ctrl_size_subgraph=30$, $\alpha=0.9$ and $\delta=0.85$.

System implementation The TS-Extractor system is implemented in browser-server architecture. Its server is implemented in Python3 and Flask, using the graph libraries *graph-tool* and *networkx* to compute relevance scores for all nodes in an input graph. SQLite is used to store graph datasets. The user interface is implemented based on Vue and Element. D3 (Bostock et al. 2011) and jQuery are used for visualization and interaction. All node-link diagrams in the views are computed and rendered using D3’s force-directed layout. The supplementary materials including the source code and a demo video are available at <https://github.com/datavis-ai/TS-Extractor>.

5 Evaluation

As an open-ended exploration tool, our system allows users to freely explore large graphs according to their own interests. How to evaluate such a system is challenging (Dörk et al. 2012). Many quantitative studies require users to complete specific predetermined tasks, which provides some comparable quantitative metrics (e.g., completion time) but constraints users' free exploration. Moreover, users often explore datasets based on their subjective interests, which makes it difficult to define quantitative metrics for user studies. To demonstrate the usability and effectiveness of the TS-Extractor system, we use two complementary ways: two case studies based on academic datasets and a user study with semi-structured interviews.

5.1 Graph datasets

We use a citation graph and a co-authorship graph as the main datasets for our case study and user study. Both datasets were extracted from AMiner (Tang et al. 2008). They have multiple node attributes, which helps to verify the effectiveness of our approach.

The citation graph consists of 8094 visualization papers (nodes) published between 1982 and 2017 and 31544 citations relationships (edges). The papers in the graph are from visualization journals including TVCG, CGF, CG&A, *Journal of Visualization*, *Information Visualization*, etc. Each paper has the same attributes, including title, authors, journal, publication date, abstract, keywords and citation counts. The directed edge from paper x to paper y indicates that x cited y .

The co-authorship graph contains 9705 authors (nodes) from four research communities including visualization (Vis), data mining (DM), human-computer interaction (HCI) and machine learning (ML). Each author has the same attributes, including name, affiliations, number of papers, citation counts, h-index and research interests. An edge exists between two authors who have co-authored or co-edited at least one paper. There are 25140 edges in the graph. The weight of each edge is the number of the corresponding co-authored papers. All co-authored papers in this dataset were published before 2013.

5.2 Case study

5.2.1 Study 1: Exploring the citation graph

We invited a first-year graduate student who is interested in graph visualization to use our system. He expected that our system could help him learn more about graph visualization. To select a related paper as his focus node, he read the details of the papers in the ranking list. The paper *Graph visualization and navigation in information visualization: A survey* (with the keyword “graph visualization”) caught his attention. He selected this paper as his first focus paper. However, for comparative analysis, he did not select any AOIs for this focus paper. As shown in Fig. 8a, the extracted subgraph is centered at the survey paper *Visual analysis of large graphs: State-of-the-art and future research challenges*. This survey paper is the most relevant neighbor of the focus paper. The big white dot at the center of the focus paper indicates that the focus paper has a large number of remaining neighbors. He added the focus paper to the matching panel and then chose “graph visualization” from the keywords of the focus paper as a matching condition to find all papers with this keyword. However, there are only seven matching papers (Fig. 8a). While adjusting the parameter *ctrl_size_subgraph* to increase the size of the subgraph, he did not find more papers with the keyword “graph visualization.”

To obtain more papers with the keyword “graph visualization,” this time he selected the “keywords” attribute of the focus paper as its AOI and re-extracted a new subgraph. If the TS-Extractor system is effective, it should be able to capture a large number of papers with the keyword “graph visualization.” He again selected the keyword “graph visualization” as a matching condition. To his surprise, all the papers in the re-extracted subgraph have the keyword “graph visualization” as shown in Fig. 8b. This means that the system can help him extract the desired subgraph. Through this comparison, he began to believe in the system. After reading the details of each paper (e.g., abstract and keywords), he summarized four interesting research topics in the field of graph visualization. The first one is interactive large graph visualization (**b1**) which allows users to explore large graphs via interactive methods (e.g., *Semantic Substrates*, *NodeTrix* and *ASK-GraphView*). The second one is edge bundling techniques (**b2**) that aim to reduce visual clutter induced

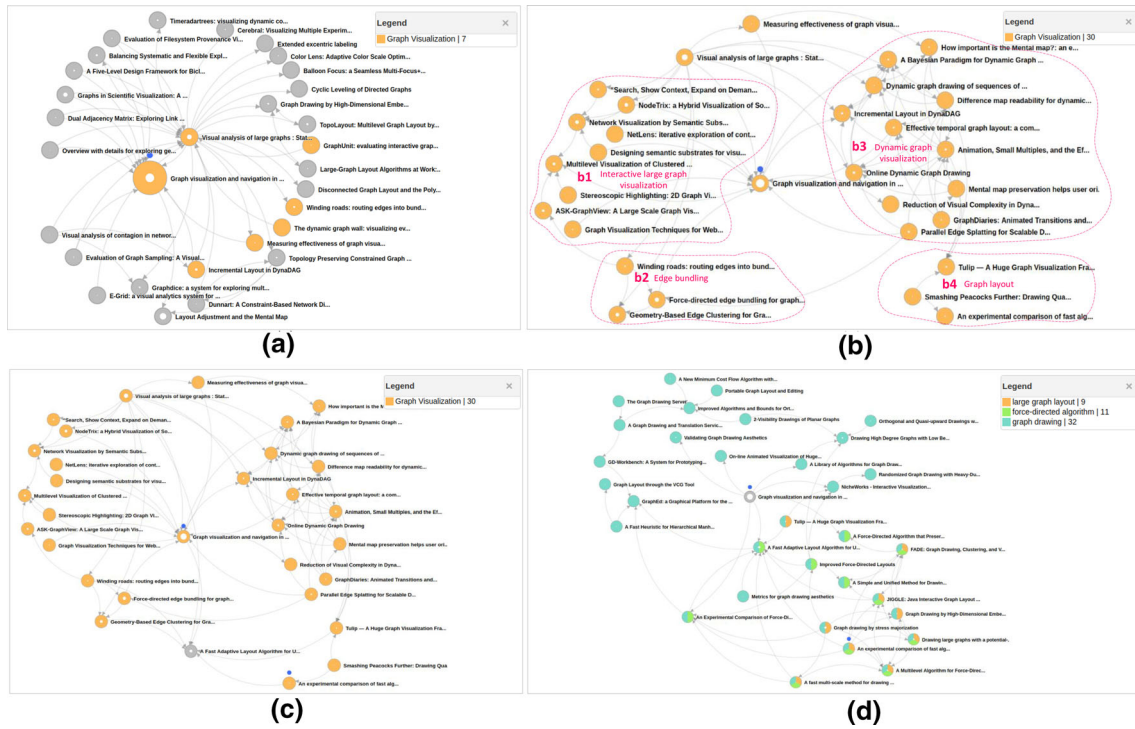


Fig. 8 Exploring graph visualization in the citation graph. **a** Our user selected a survey paper with the keyword “graph visualization” from the ranking list as his first focus paper (with a blue dot on its top). But he did not consider its any attributes. The extracted subgraph only contains seven papers with the keyword “graph visualization.” **b** After he considered the “keywords” attribute of the first focus paper, the re-extracted subgraph contains 30 papers with “graph visualization.” He selected a paper with the keywords “graph visualization,” “large graph layout,” “force-directed algorithm” and “graph drawing” from **b4** as his second focus paper. **c** When he did not consider any attributes of the second focus paper, the subgraph extracted based on the two focus papers only adds a paper (in grey) on the basis of (**b**). **d** After he also considered the “keywords” attribute of the second focus paper, the re-extracted subgraph contains nine papers with “large graph layout,” 11 papers with “force-directed algorithm” and 32 papers with “graph drawing”

by edge crossings. The third one is dynamic graph visualization (**b3**) whose goal is to lay out graphs that change over time. The last one is graph layout or graph drawing (**b4**) which provides automatic algorithms to draw the entire graph. This knowledge helped him further his understanding of the field of graph visualization.

To further understand graph layout, he selected the paper *An experimental comparison of fast algorithms for drawing general large graphs* (with the keywords “graph visualization,” “large graph layout,” “force-directed algorithm” and “graph drawing”) from **b4** as his second focus paper. For further comparative analysis, he did not consider any attributes of the second focus node. The subgraph extracted based on the two focus papers is shown in Fig. 8c. To his disappointment, this subgraph only adds one paper (i.e., *A Fast Adaptive Layout Algorithm for Undirected Graphs*) on the basis of the previous subgraph (Fig. 8b). Then, he also selected the “keywords” attribute of the second focus paper as its AOI. The re-extracted subgraph is shown in Fig. 8d. Through the matching panel, he found nine papers with the keyword “large graph layout” and 11 papers with the keyword “force-directed algorithm.” There are six papers with both “large graph layout” and “force-directed algorithm.” Almost all papers have the keyword “graph drawing.” Reading these related papers could help him understand the overall overview of graph layout. Through this comparison, he believed that our system was effective and useful.

5.2.2 Study 2: Exploring the co-authorship graph

In Study 1, our user gradually chose his focus nodes and attributes of interest to extract relevant nodes and discover new knowledge. In this case study, we want to demonstrate that when multiple focus nodes are selected simultaneously, considering the node attributes of each focus node can help the user extract

corresponding semantic node groups and discover interesting patterns. We simultaneously select three prestigious Vis authors as our focus authors: *Thomas Ertl* at the *University of Stuttgart*, *Kwan-Liu Ma* at the *University of California at Davis* and *John T. Stasko* at the *Georgia Institute of Technology*.

We first do not consider any attributes of each focus author. As shown in Fig. 9a, the extracted subgraph mainly contains nodes that are structurally relevant to the focus authors. Authors at the same institution as a focus author are highlighted in the corresponding color (see legend). There are strong intra-institutional collaborations within the University of Stuttgart. In addition to intra-institutional collaborations, each focus author has actively participated in cross-institutional collaborations. For example, *John T. Stasko* has collaborated with researchers at Microsoft Research (*G. Robertson* and *M. Czerwinski*). *Thomas Ertl* has not established collaborations with *John T. Stasko* and *Kwan-Liu Ma*. But both *Thomas Ertl* and *Kwan-Liu Ma* have collaborated with *Min Chen*. The author *Min Chen*, a Vis researcher at the *University of Oxford*, is an important bridging node connecting European Vis researchers and American Vis researchers. We then bring in his five neighbors with high paper count. The five neighbors are placed in a vertical layout and ranked by their paper count. As shown in Fig. 9a, several interesting Vis authors are added in the current subgraph. *Daniel Weiskopf* is another excellent Vis researcher at the University of Stuttgart. *David S. Ebert*, a Vis researcher at Purdue University, is another bridging node.

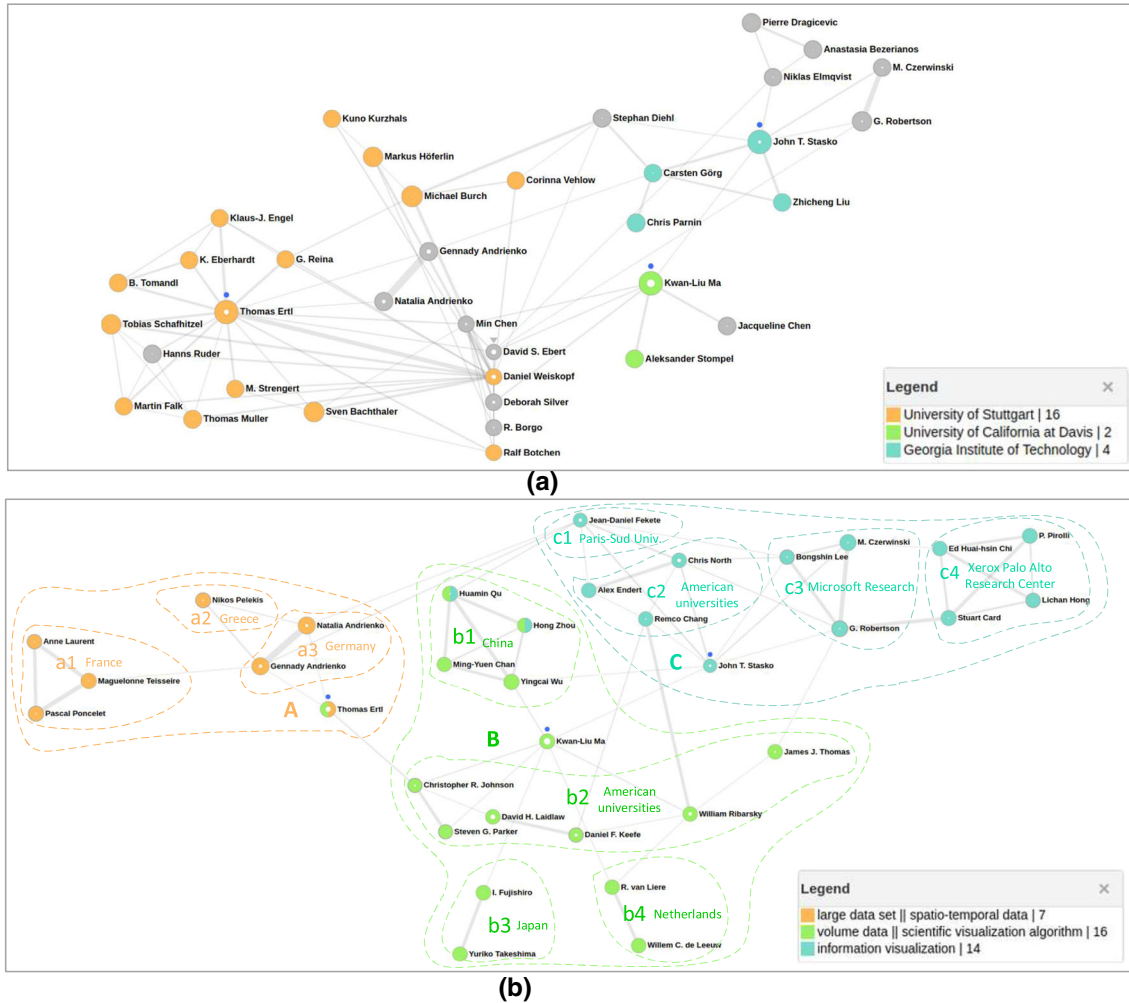


Fig. 9 Exploring the co-authorship graph with *Thomas Ertl*, *Kwan-Liu Ma* and *John T. Stasko* as focus authors. **a** Extracting a subgraph without considering any attributes of each focus author and adding five neighbors of the author *Min Chen* based on their paper count. **c** Re-extracting a new subgraph after considering the “research interests” attribute of each focus author. The subgraph is divided into three clusters (A, B and C). The authors in each cluster are from different countries or institutions but share the same/similar research interests (e.g., “information visualization” in C)

Next, to obtain more authors sharing the same/similar research interests with the three focus authors, we select the “research interests” attribute as the AOI of each focus author. As shown in Fig. 9b, the re-extracted subgraph is divided into three clusters (**A**, **B** and **C**) and the authors in each cluster have the same/similar research interests (see legend).

The authors in **A** are captured based on the research interest “large dataset” or “spatiotemporal data” of *Thomas Ertl*. They come from France (**a1**), Greece (**a2**) and Germany (**a3**). However, not all of these authors are Vis researchers. *Anne Laurent* et al. (**a1**) are DM researchers who develop data mining algorithms on large datasets. *Nikos Pelekis* (**a2**) is a DM researcher who mines movement data. *Natalia Andrienko* and *Gennady Andrienko* (**a3**) are Vis researchers who focus on visualization of spatiotemporal data. Collaboration between Vis researchers and DM researchers shows that these two fields are very related. Visualization can be regarded as a visual data mining technique.

The authors in **B** are induced by the research interest “scientific visualization” or “volume data” of *Kwan-Liu Ma*. They are Vis researchers from China (**b1**), American universities such as the University of Utah and Brown University (**b2**), Japan (**b3**) and Netherlands (**b4**). From the cluster, it can be inferred that visualization is booming in the USA. American universities have more Vis researchers than universities in other countries. In addition, it is encouraging to see that visualization is growing in China.

The authors in **C** are captured based on the research interest “information visualization” of *John T. Stasko*. They are Vis researchers from Paris-Sud University (**c1**), American universities such as Tufts University and the University of Maryland (**c2**), Microsoft Research (**c3**) and Xerox Palo Alto Research Center (**c4**). From this cluster, it can be inferred that in the field of visualization, collaboration between academia and industry is widespread. Moreover, we can also find the collaboration between American Vis researchers and European Vis researchers.

Through the above comparative analysis, we think that considering attributes of the focus nodes can help us obtain more insights on the same dataset.

5.3 User study

We conducted a user study with semi-structured interviews to collect qualitative feedback on TS-Extractor. We recruited five graduate students (ages 22–26) engaged in visualization research as participants (P1–P5). P1–P3 (male) work on graph visualization. P4 (female) and P5 (male) focus on visual explanation of deep learning models. In the user study, we provided two datasets: the citation graph and the co-authorship graph. Each participant can choose one of interest to start his/her exploration.

5.3.1 Procedure

Due to the impact of COVID-19, the user study was conducted through Web conferencing. First, we provided the participants with a 20-minute tutorial to explain the TS-Extractor system and demonstrate how to use it to explore large graphs. Then, each participant was asked to perform the following four general tasks:

1. Open-ended graph exploration: Each participant was allowed to freely explore the selected dataset according to his/her own interests. This task could help every participant be familiar with each module of the system, understand TS-Extractor’s pipeline and make sense of the system’s visualization and interaction, so that they could give us more detailed feedbacks in semi-structured interviews.
2. Investigating the effect of parameters on subgraph extraction: Parameters α and δ are involved in the relevance computation. The participants were asked to investigate the effect of these two parameters on subgraph quality. We wanted to know how each parameter affected the extracted results, whether the extracted results were sensitive to the parameters, and what parameter values would extract satisfactory results for participants. We also asked the participants to explore how large a subgraph was sufficient for them by adjusting the parameter *ctrl_size_subgraph*.
3. Comparing the subgraphs extracted in two cases: ignoring any attributes of the focus nodes and considering AOIs of the focus nodes. If a participant extracted a subgraph by considering AOIs of the focus nodes, we asked him/her to compare it with the subgraph re-extracted without considering any attributes of the focus nodes, and vice versa. This task was used to verify whether TS-Extractor could capture more nodes sharing similar attribute values to the focus nodes when considering the focus nodes’ AOIs. In addition, we would like to know whether the extracted subgraphs match the users’ subjective interests.

4. Expanding the extracted subgraphs by adding remaining neighbors of nodes of interest: With this task, we could investigate which layout (circular layout or vertical layout) participants preferred and whether newly added neighbors could provide interesting information.

In particular, we asked each participant to do Task 2 and Task 3 on both datasets provided so that we could investigate the generalizability of the system. We did not have a time limit for each task, so that each participant could think more about each exploration process. After approximately 2.5 h, every participant completed the above four tasks. Next, we conducted a semi-structured interview with each participant. Each interview session was guided by questions based on the above tasks and lasted approximately 20–30 min. We recorded the participants' responses and took note of their feedbacks.

5.3.2 Result

In the following, we summarize and discuss the results collected from the semi-structured interviews.

The system's visualization and interactive design: In terms of overall impression of the system, all participants stated that the system was easy to learn and use. In each interview session, the respondent commented that the ranking list could help him/her quickly select focus nodes in open-end exploration. P4 and P5 reported that the search result view could help them quickly locate nodes of interest from many search results. Participants were satisfied with the matching panel and the visual design of nodes. For example, P1 stated “*I really like the matching panel. It can help me find groups of nodes with similar attribute values in the subgraph. The visual coding of the remaining neighbors of each node is useful. It helps me perceive the implied context of each node.*” For the history view, participants commented that it was useful when they compared subgraphs, but at other times they did not need it. In addition, P3 added that it would be very useful if the user wanted to restore a subgraph immediately.

The effect of parameters on subgraph extraction: All participants agreed that the extracted results were not sensitive to either α or δ in each dataset. It was even possible to extract the same subgraph within a wide range of parameter values. An example can be given. P2 selected the paper *Empirical Studies in Information Visualization: Seven Scenarios* (with the keyword “information visualization”) as his focus node and considered the “keywords” attribute of this paper. Then, he extracted the same subgraph when $\alpha=(0.43, 1]$ and $\delta=(0.57, 1]$. He was satisfied with the subgraph because it contained a large number of papers with the keyword “information visualization.” Most participants reported that increasing α helped to extract subgraphs containing more nodes similar to focus nodes. In addition, increasing δ could help capture nodes sharing similar attribute values with focus nodes or nodes connected by edges with large weights. The participants believed that the default parameter values ($\alpha=0.9$, $\delta=0.85$) were applied to many analysis scenarios in both datasets. As for the appropriate subgraph size, most participants stated that too many nodes would increase the burden of their exploration. They suggested that the number of nodes should be controlled below 100.

The effect of node attributes on subgraph extraction: Through comparative analysis, all participants appreciated that the system really could capture nodes sharing the same/similar attribute values with the focus nodes when considering the attributes of the focus nodes. Three comparison cases made by participants are presented:

1. P1 first picked the paper *Multi-Level Graph Layout on the GPU* (with the keyword “graph drawing” and 121 citations) as his focus node but did not consider its attributes. In extracted subgraph, he found 27 papers with the keyword “graph drawing” and 14 papers with no less than 121 citations. Then, he considered the “keywords” attribute and the “num_citation” attribute of his focus node and re-extracted a subgraph. This time he found 44 papers with the keyword “graph drawing” and 21 papers with no less than 121 citations in the subgraph. In this case, his parameter values were $ctrl_size_subgraph=44$, $\alpha=0.9$, $\delta=0.85$.
2. P4 first selected the paper *Machine Learning to Boost the Next Generation of Visualization Technology* (with the keywords “machine learning,” “information visualization” and “scientific visualization”) as her focus node but did not consider any attributes. In the extracted subgraph, she only found five papers with the keywords “machine learning,” “information visualization” or “scientific visualization.” After considering the “keywords” attribute of the focus node, she was surprised to find 30 papers with the keywords “machine learning,” “information visualization” or “scientific visualization” in the re-extracted subgraph. In the case, her parameter values were $ctrl_size_subgraph=30$, $\alpha=0.9$, and $\delta=0.85$.

3. P3 picked the authors *Kwan-Liu Ma* (with the research interests “volume data,” “scientific visualization algorithm,” “visual analytics” and “human–computer interaction”) and *Huamin Qu* (with the research interests “real volume data,” “complex data” and “information visualization”) as his focus nodes. When the attributes of both focus nodes were not considered, he only found nine authors with similar research interests to *Kwan-Liu Ma* and six authors with similar research interests to *Huamin Qu* in the extracted subgraph. When he considered the “research interests” attribute of the two focus nodes, he could find 16 authors with similar research interests to *Kwan-Liu Ma* and 13 authors with similar research interests to *Huamin Qu* in the re-extracted subgraph. In this case, his parameter values were $ctrl_size_subgraph=30$, $\alpha=0.9$, and $\delta=0.85$.

In addition, the participants stated that considering the attributes of the focus nodes could make the semantics of the extracted subgraphs clearer and help facilitate their exploration.

Subgraph expansion: Participants were not consistent in their choices for the layout of the newly added neighbors. Three participants (P2, P3 and P4) preferred vertical layout, while two participants (P1 and P5) preferred circular layout. P4 commented that the vertical layout could help her intuitively perceive how the neighbors’ attribute values (e.g., number of citations, number of papers, or h-index) were ranked. P5 stated that the circular layout could save vertical screen space, especially when many neighbors were added. Most participants agreed that the added neighbors could help them explore the contextual information of nodes of interest such as discovering hidden important citations or co-authors.

6 Discussion

To capture nodes sharing the same/similar attribute values with focus nodes, a simple method is to calculate a match score for each node in G_{in} based on the AOI values (or KOIs) of focus nodes. However, we will encounter a problem. Assume there are M focus nodes in the focus set and each focus node has N AOI values. If the AOI values of all focus nodes are completely different, there are $M \times N$ AOI values. We then need to compute $M \times N$ match scores for each node in G_{in} based on the $M \times N$ AOI values. How do we compute the final match score of a node based on its $M \times N$ match scores? In this paper, we combine the augmented graph strategy and Personalized PageRank to address this problem. Personalized PageRank can compute the relevance between a set of starting nodes and all the other nodes in a large graph. Our approach converts matching into relevance computation.

In the constructed augmented graph, we simply treat all matching edges equally and assign the same weight value to each one. Our experiments show that the results are not highly sensitive to the exact choices of the weight value. However, different types of matching edges may need to be assigned different weight values. For this complex situation, how to choose the appropriate weight value for each type of matching edge is a challenge. The simplest method is to allow users to specify a weight value for each type of matching edge. But this will increase users’ burden of network exploration. One potential solution would be to automatically adjust the corresponding weight values according to the user’s exploration history. For example, matching edges corresponding to attributes that users often select have higher weight values.

A KOI is usually an attribute value of a focus node, such as *graph visualization*. TS-Extractor can help us capture nodes matching the KOIs when we use KOIs to search for focus nodes, which is similar to considering AOIs of focus nodes. The TS-Extractor system allows users to decide whether to select AOIs for each focus node. On the one hand, this enhances the flexibility and interpretability of TS-Extractor. On the other hand, this can help users compare the results of ignoring attributes and the results of considering attributes.

In addition to the two datasets used in the case studies and the user study, our system can run on an Amazon co-purchasing graph with 23,650 nodes and three attributes. However, when we try to run larger datasets (e.g., 50,000 nodes), our system runs very slowly. The calculation time of the system is spent mainly on the relevance computation. The time of the relevance computation is mainly consumed in the construction of the augmented graph G_{augm} and the computation of PPR. To find matching nodes for each keyword or attribute value, we traverse all nodes in the input graph G_{in} online. Thus, the construction time of G_{augm} increases as the number of keywords or attribute values increases. To speed up the construction of G_{augm} , we plan to create an offline “attribute_value-node_IDs” database table for all non-numeric attributes. Specifically, each row in this table stores an attribute value and IDs of nodes matching this attribute value. From this table, we can quickly find the corresponding matching nodes for KOIs or values of AOIs during

the construction of G_{augm} . Similarly, the computation time of PPR increases with the size of G_{augm} . To speed up PPR, we plan to implement a fast PPR (Tong et al. 2006).

Our approach can support both non-numeric and numeric attributes. In fact, the type of attribute only affects how the augmented graph G_{augm} is constructed. For common non-numeric attributes (e.g., the “keywords” attribute of a paper and the “research interests” attribute of an author), we construct G_{augm} through the similarity matching of attribute values. However, for numeric attributes (e.g., the “citation count” attribute of a paper and the “paper count” attribute of an author), it is not appropriate to use similarity matching of attribute values to construct G_{augm} . In our system, we construct G_{augm} by comparison of attribute values. Assume we select a paper from the citation graph as our focus node and its citation count is 30. If we consider the “citation count” attribute of this focus node, our system will help us capture papers with no less than 30 citations from the citation graph. Different ways to construct G_{augm} will lead to different results. To make our system more flexible, we plan to allow users to define a way to construct G_{augm} for any numeric attribute.

7 Conclusion

In this paper, we introduce TS-Extractor, a local graph exploration approach that can combine graph topology and semantic information (e.g., node attributes) to extract a relevant subgraph with clear semantics according to the user’s interest. We propose a relevance computation method that computes a relevance score for each node in the input graph by constructing an augmented graph using node attributes. To capture more nodes sharing the same/similar attribute values with the focus nodes, we introduce a diffusion method. In addition, we develop the TS-Extractor system that provides visualization and interaction designs to enable the user to explore large graphs from a local perspective. Our case studies and user study on real-world graph datasets demonstrate the usability and effectiveness of TS-Extractor.

Acknowledgements This research has been supported by the National Natural Science Foundation of China (61725105).

References

- Abello J, Van Ham F, Krishnan N (2006) Ask-graphview: a large scale graph visualization system. *IEEE Trans Visual Comput Graph* 12(5):669–676
- Abello J, Hadlak S, Schumann H, Schulz HJ (2014) A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Trans Visual Comput Graph* 20(3):337–350
- Auber D, Chiricota Y, Jourdan F, Melançon G (2003) Multiscale visualization of small world networks. In: *IEEE symposium on information visualization 2003* (IEEE Cat. No. 03TH8714), IEEE, pp 75–81
- Bostock M, Ogievetsky V, Heer J (2011) D³ data-driven documents. *IEEE Trans Visual Comput Graph* 17(12):2301–2309
- Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Networks ISDN Syst* 30(1–7):107–117
- Chau DH, Kittur A, Hong JI, Faloutsos C (2011) Apollo: Interactive large graph sensemaking by combining machine learning and visualization. In: *Acm Sigkdd international conference on knowledge discovery & data mining*
- Chen W, Guo F, Han D, Pan J, Nie X, Xia J, Zhang X (2019) Structure-based suggestive exploration: a new approach for effective exploration of large networks. *IEEE Trans Visual Comput Graph* 25(1):555–565
- Crnovrsanin T, Liao I, Wuy Y, Ma KL (2011) Visual recommendations for network navigation. In: *Eurographics*
- Dörk M, Riche NH, Ramos G, Dumais S (2012) Pivotpaths: strolling through faceted information spaces. *IEEE Trans Visual Comput Graph* 18(12):2709–2718
- Dunne C, Shneiderman B (2013) Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In: *Sigchi conference on human factors in computing systems*
- Dupont P (2006) Relevant subgraph extraction from random walks in a graph. *Res Rep Rr* 13(4):264–268
- Edge D, Larson J, Mobius M, White C (2018) Trimming the hairball: Edge cutting strategies for making dense graphs usable. In: *2018 IEEE international conference on Big Data (Big Data)*. IEEE, pp 3951–3958
- Frank VH, Adam P (2009) “search, show context, expand on demand”: supporting large graph exploration with degree-of-interest. *IEEE Trans Visual Comput Graph* 15(6):953
- Furnas GW (1986) Generalized fisheye views 17(4)
- Ghoniem M, Mcgee F, Melançon G, Otjacques B, Pinaud B (2019) The state of the art in multilayer network visualization. *arXiv preprint arXiv:1902.06815*
- Haveliwala TH (2003) Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. Technical Report 2003-29, Stanford InfoLab, <http://ilpubs.stanford.edu:8090/750/>, extended version of the WWW2002 paper on Topic-Sensitive PageRank
- Heer J, Boyd D (2005) Vizster: visualizing online social networks. In: *IEEE symposium on information visualization, 2005. INFOVIS 2005*. IEEE, pp 32–39

- Herman I, Melançon G, Marshall MS (2002) Graph visualization and navigation in information visualization: a survey. *IEEE Trans Visual Comput Graph* 6(1):24–43
- Holten D (2006) Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Trans Visual Comput Graph* 12(5):741–748
- Hong Z, Xu P, Yuan X, Qu H (2013) Edge bundling in information visualization. *Tsinghua Sci Technol* 18(2):145–156
- Jia Y, Hoberock J, Garland M, Hart J (2008) On the visualization of social and other scale-free networks. *IEEE Trans Visual Comput Graph* 14(6):1285–1292
- Kairam S, Riche NH, Drucker S, Fernandez R, Heer J (2015) Refinery: visual exploration of large, heterogeneous networks through associative browsing. *Comput Graph Forum Wiley Online Library* 34:301–310
- Laumond A, Melançon G, Pinaud B (2017) edoi: Exploratory degree of interest exploration of multilayer networks based on user interest. In: *VIS 2017*, Poster session
- Leskovec J, Faloutsos C (2006) Sampling from large graphs. In: *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp 631–636
- Li C, Baciú G, Wang Y (2017) Module-based visualization of large-scale graph network data. *J Visual* 20(2):205–215
- Liu S, Cui W, Wu Y, Liu M (2014) A survey on information visualization: recent advances and challenges. *Visual Comput* 30(12):1373–1393
- Moscovich T, Chevalier F, Henry N, Pietriga E, Fekete JD (2009) Topology-aware navigation in large networks. In: *Sigchi conference on human factors in computing systems*
- Pienta R, Abello J, Kahng M, Chau DH (2015) Scalable graph exploration and visualization: Sensemaking challenges and opportunities. In: *International conference on Big Data & smart computing*
- Pienta R, Kahng M, Lin Z, Vreeken J, Talukdar P, Abello J, Parameswaran G, Chau DH (2017) Facets: adaptive local exploration of large graphs. In: *Proceedings of the 2017 SIAM international conference on Data Mining*. SIAM, pp 597–605
- Pretorius AJ, Wijk JJV (2008) Visual inspection of multivariate graphs
- Shen Z, Ma KL, Eliassi-Rad T (2006) Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Trans Visual Comput Graph* 12(6):1427–1439
- Shi L, Liao Q, Tong H, Hu Y, Zhao Y, Lin C (2014) Hierarchical focus+ context heterogeneous network visualization. In: *2014 IEEE Pacific visualization symposium (PacificVis)*. IEEE, pp 89–96
- Shneiderman B, Aris A (2006) Network visualization by semantic substrates. *IEEE Trans Visual Comput Graph* 12(5):733–740
- Stef VDE, Wijk V, Jarke J (2014) Multivariate network exploration and presentation: from detail to overview via selections and aggregations. *IEEE Trans Visual Comput Graph* 20(12):2310
- Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: extraction and mining of academic social networks. In: *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp 990–998
- Tong H, Faloutsos C, Pan JY (2006) Fast random walk with restart and its applications. In: *Sixth international conference on data mining (ICDM'06)*. IEEE, pp 613–622
- Von Landesberger T, Kuijper A, Schreck T, Kohlhammer J, van Wijk JJ, Fekete JD, Fellner DW (2011) Visual analysis of large graphs: state-of-the-art and future research challenges. *Comput Graph Forum Wiley Online Library* 30:1719–1749
- Wattenberg M (2006) Visual exploration of multivariate graphs. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, pp 811–819
- Zhan C, Zhang D, Wang Y, Lin D, Wang H (2019) Ies-backbone: an interactive edge selection based backbone method for small world network visualization. *IEEE Access* PP(99):1
- Zhao Y, Luo X, Lin X, Wang H, Chen W (2019) Visual analytics for electromagnetic situation awareness in radio monitoring and management. *IEEE Trans Visual Comput Graph* PP(99):1