



Lijing Lin · Jiacheng Yu · Fan Hong · Chufan Lai · Siming Chen · Xiaoru Yuan

GraphInterpreter: a visual analytics approach for dynamic networks evolution exploration via topic models

Received: 25 December 2023 / Accepted: 20 February 2024 / Published online: 22 May 2024
© The Visualization Society of Japan 2024

Abstract We propose a novel visual analytics approach based on the Latent Dirichlet Allocation (LDA) model for exploring and interpreting the dynamic evolution of networks. In this approach, we define networks as documents and relationships within networks as words. Using this definition, the LDA model is able to extract a list of structures that fuse relationships and connect the network features. We project networks described by the extracted structures with probabilistic assignments as points into a two-dimensional space via dimensionality reduction techniques. Users can identify evolution states in dynamic networks, including stable states, recurrent states, outlier states, and state transitions. To facilitate the interpretation of evolution states, we provide a novel small multiples view that shows how the extracted structures behave over time. We demonstrate the effectiveness of our work through case studies conducted on two real-world dynamic networks.

Keywords Visual analytics · Dynamic networks · Topic models

L. Lin · F. Hong · C. Lai · X. Yuan (✉)
Key Laboratory of Machine Perception (Ministry of Education), School of Intelligence Science and Technology, Peking University, Beijing, China
E-mail: xiaoru.yuan@pku.edu.cn

L. Lin
E-mail: lijinglin29@gmail.com

F. Hong
E-mail: fan.hong@pku.edu.cn

C. Lai
E-mail: chufan.lai.1990@gmail.com

L. Lin · J. Yu · F. Hong · C. Lai · X. Yuan
National Engineering Laboratory for Big Data Analysis and Application, Peking University, Beijing, China
E-mail: jiachengyu@pku.edu.cn

J. Yu
Center for Data Science, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China

S. Chen
School of Data Science, Fudan University, Shanghai, China
E-mail: simingchen3@gmail.com

1 Introduction

Dynamic network is a series of networks describing how the entities and their relationships change over time. It has been used to model a wide range of scenarios, such as the information diffusion in social media, financial transactions, and academic collaboration changes. In dynamic network analysis, summarizing how the dynamic network evolves and understanding why the networks have such evolved patterns are two important challenges. However, networks always evolve over a long period of time, with complex topological changes, which impedes exploration and interpretation of dynamic network evolution.

In dynamic network visualization, the two most commonly used methods are small multiples and animation. The small multiples techniques, mapping time steps to the screen space, visualize a sequence of networks as static snapshots, while animation techniques, mapping time steps to time, show the networks one by one with a topological transition. For each network at a given time step, both methods show the graph structure directly. However, to identify different states in dynamic network, such as stable states and recurrent states, users must take effort to discover similarities and differences among networks by relating and comparing them. When the number of time steps is large and the topological structures are complex, relating and comparing networks that are far apart from each other could lead to high cognitive load and significant time cost.

A general approach to detecting states in dynamic networks is to treat networks at each time step as a high-dimensional vector and then to project them as points in a lower-dimensional space via dimensionality reduction techniques (van den Elzen et al. 2016). Points that are near to each other indicate that the corresponding networks are similar. However, interpreting why some networks are similar or different is difficult, because the detailed information of networks is highly aggregated and omitted in the dimensionality reduction results.

To facilitate the dynamic network evolution exploration and interpretation, we propose a novel visual analytics method based on Latent Dirichlet Allocation (LDA) model, a widely used topic model in text analysis. The LDA topic model explains the similarities of documents from the unobserved latent topics instead of low-level words.

In our approach, we define network at each time step as a document and relationships between entities (links) as words, respectively.

The inclusion information between networks and relationships is used as the input of an LDA model. The outputs are a list of topics fused from relationships, which we denote as extracted structures or structures for short. Simultaneously, networks are described by these extracted structures with probabilistic assignments. In other words, the extracted structures act as high-level bridges connecting a sequence of networks and their relationships. Therefore, we could summarize the evolution of dynamic network via the description of networks based on the extracted structures and interpret the evolution patterns through the description of extracted structures based on the relationships.

Built upon the LDA-based analysis, GraphInterpreter is a novel visual analytics system designed to assist users in exploring and interpreting dynamic network evolution. First, GraphInterpreter provides the summary of the evolution of the dynamic network for users by projecting networks described by extracted structures as points into a two-dimensional space via dimensionality reduction techniques. Then users are supported to identify different states during network evolution regarding structures, as the distance measurements are taken from the extracted structures contained in networks. GraphInterpreter also includes a novel small multiples view that displays the topology of extracted structures and their changes over time, for the users to reveal the semantic implications of structures to facilitate the interpretation of the evolution.

Our contributions could be summarized as

- We introduce a novel dynamic network evolution analysis approach based on topic models, with a compact layout algorithm-based honeycomb to reduce visual clutter and a strategy to differentiate nodes to emphasize the topology changes.
- We present a novel visual analytics system for dynamic network evolution exploration and interpretation by reducing networks into a two-dimensional space and providing novel small multiples to show the topology changes from the aspects of extracted structures.
- We demonstrate the effectiveness of our proposed approach and visual analytics system by conducting experiments on two different datasets.

2 Related work

We review the existing works that are most related to our method, including techniques developed for dynamic network visualization and visual analysis, and topic modeling in graph visualization.

2.1 Dynamic network visualization

Dynamic network visualization focuses on representing relationships between entities which change over time. The two dominant visualization methods are animation and small multiples (Beck et al. 2014, 2016).

The animation techniques show networks in each time step one by one like a movie, e.g., (Huang et al. 1998; Erten et al. 2003a; Rufiange and McGuffin 2013). In these approaches, to analyze the evolution of dynamic network, users need to focus on many moving glyphs simultaneously and to keep tracks of various changes over time, which leads to high cognitive load and costs lots of time. To reduce users' burden, animation technique is always closely related to maintaining consistent layouts for mental map preservation, such as (Forrester et al. 2004; Bach et al. 2014). However, preserving mental map is challenging, as maintaining the position of nodes would influence esthetic quality of layouts (Gorochowski et al. 2012).

The small multiples techniques visualize a sequence of networks as static snapshots by juxtaposition (Burch et al. 2010), superimposition (Dwyer and Eades 2002; Erten et al. 2003b), or the integration of these two methods (Shi et al. 2011; van den Elzen et al. 2016). These approaches show an overview of a dynamic network. However, mapping time to space would reduce the space for each snapshot, which decreases the readability of individual snapshots.

To deal with the space limitation, various novel visualizations have been proposed. For example, TimeArcTrees (Greilich et al. 2009) maps time steps as vertical axes and arranges nodes on them. Time-Arcs (Dang et al. 2016) visualizes entities as lines and the relationships as arcs among these lines. These approaches could provide a compact overview but lose the readability compared with node-link diagram. Also, in small multiples approaches, it is difficult for users to relate networks in each time step, which might be far apart from each other, to discover patterns. To make the relating convenient, maintaining the stability of layouts is important. Che et al. (2015) use a Laplacian constrained distance embedding method to maintain the overall structures of a sequence of graphs. Wang et al. (2017) present an improved stress majorization method to constrain node positions by considering their updating directions and distances during iterations.

In our work, we use small multiples to show snapshots. As understanding the meaning of snapshots needs lots of comparison and exploration, the animation is not suitable here. To ease users' burden in small multiples approach, we calculate a supergraph layout as an overview and use this layout in layout snapshots for mental map preservation (Diehl et al. 2001; Diehl and Görg 2002). We also provide multiple interactions, such as brushing, merging, and splitting, and novel glyphs to help the exploration and comparison.

2.2 Dynamic network visual analysis

Dynamic network visual analysis focuses on analyzing the evolution of relationships between entities. Various visual analysis techniques have been proposed to analyze the evolution of subgraphs or the whole networks. Subgraph evolution analysis focuses on the extraction and visualization of the subgraphs. To analyze subgraphs with similar trends over time, Hadlak et al. (2013) and Steiger et al. (2014) group nodes and edges according to their time-varying attributes. Different from these methods, which only consider attributes, some techniques are designed to mine and visualize the evolution of communities in dynamic networks based on the topological structures of networks, e.g., (Falkowski et al. 2006; Vehlow et al. 2015, 2016). While these methods are all based on predefined rules, Abello et al. (2014) introduce a modular degree-of-interest specification to extract user-defined dynamic patterns.

In contrast to these techniques, our goal is to analyze the evolution of the whole network instead of the subgraphs. In this aspect, there are two similar works: von Landesberger et al. (2015) use the dimensionality reduction method to analyze the evolution of the contagion process with node attributes, and van den Elzen et al. (2016) consider both topology information and related attributes to project graphs into two-dimensional space. Both methods calculate high-dimensional vectors describing graphs with raw data, which would be very large and sparse. Compared with them, we use higher-level information to define the related vectors. What is more, our work provides a novel visual analytics system supporting users to understand the evolution of dynamic networks, not just showing the raw data.

2.3 Topic modeling in graph visualization

Topic modeling, such as Latent Dirichlet Allocation (Blei et al. 2002), is a widely used machine learning algorithm for finding patterns of text corpora based on the distribution of topics. Topic modeling has been extended to detect communities in networks (Zhang et al. 2007; Henderson and Eliassi-Rad 2009; Cha and Cho 2012). In these works, they propose to treat each node in the network as a document and neighboring nodes as words. Extracted nodes with high probabilities in each topic are considered as a community. Different from these methods, in our work, networks at each time step are considered as documents and corresponding relationships are words. Therefore, extracted high-probability relationships in each topic form a semantic structure that could be used to summarize networks with high probabilities of this topic. To the best of our knowledge, our method is the first to use topic modeling in analyzing the evolution of dynamic networks.

3 Overview

In this section, we first present our motivation and analysis tasks of dynamic network evolution visual analysis. Afterward, we discuss our approach and how to apply the topic models to dynamic network. Finally, we describe three main steps in our visual analytics approach: vectorization of dynamic network, calculation via topic models, and projection for network evolution.

3.1 Analysis tasks

To explore and understand the evolution of dynamic network, it is critical to answer the following two questions:

- How does the dynamic network evolve?
- How to understand such evolution patterns?

To address these two questions, traditional visualization techniques, such as small multiples and animation, require users to relate and compare multiple networks which may be far apart either in the screen space or in the play time. In small multiples techniques, it is difficult for users to summarize the evolution from a large amount of changing entities and relationships across a long sequence of snapshots. Meanwhile, in animation techniques, it is very time-consuming to derive the evolution patterns by tracking and relating a large number of changing entities and relationships simultaneously. Therefore, both small multiples and animation techniques could lead to a high cognitive load and significant time cost when dealing with these two questions.

To support the extraction, exploration, and interpretation of the dynamic network evolution, our analysis tasks include

- Identity the evolution patterns, including stable states, recurrent states, outlier states, and state transitions.
- Interpret the evolution patterns, e.g., which part of some networks are similar or different.

3.2 Rationales

To fulfill our analysis tasks, it is necessary to extract semantic structures from networks for analysis and to interpret the dynamic network evolution. Two traditional approaches for structure extraction are clustering and community detection.

- **Clustering** It is a common approach to divide entities into several clusters based on the similarities of their attributes, such as K-Means and hierarchical clustering. The most important property of the clustering results is that entities in the same cluster are more similar to each other than those in different clusters. In other words, the clustering techniques focus on grouping similar entities together.
- **Community detection** The main task of community detection approaches is to identify and group entities with dense connections. Entities in one community are more densely connected internally than with the rest of the network.

As discussed, traditional approaches could not support the extraction of evolution patterns in the dynamic network, because they focus on entities and relationships in a static network. The relations among networks of different time steps are much more complicated than clusters or communities. The structures shared by different networks and the evolution of structures are usually too diverse to be defined in prior. As a result, we need more sophisticated methods to fulfill our analysis tasks. In text analysis, the LDA model is used to extract latent topics that explain why documents are similar in the higher topic level instead of the word level. The extraction of the latent topics is based on the inclusion relations between documents and words, rather than predefined. This situation is very similar with dynamic network analysis that predefined features cannot satisfy the requirements of analysis. Inspired by the idea of topic modeling, we employ LDA model to extract latent topics in dynamic networks to analyze the similarities and differences between networks of different time steps, and to explore their evolution. We refer to these latent topics as structures in our scenarios (Fig. 1).

However, the application of the LDA model to the dynamic network is a critical problem. There are three main types of objects in dynamic network, named entities, links, and networks in different time steps. Some existing works (Zhang et al. 2007; Henderson and Eliassi-Rad 2009; Cha and Cho 2012) have utilized the LDA model for graph analysis. They have focused only on fuzzy entity clustering in static networks, rather than evolution analysis in dynamic networks. Therefore, our goal is distinct from theirs.

We propose a new utilization of the LDA model for dynamic network visualization. As shown in Fig. 2, we illustrate how we define *networks* at different time steps as *documents* and *links* in networks as *words*. Using this definition, the LDA model is able to extract a list of structures fused from links, with each structure being treated as a probability distribution over all links. Furthermore, networks at different time steps are assigned to structures with probabilities that indicate how much of the structures are identified in each network. In short, the extracted structures serve as connectors between the networks and links.

3.3 Visual analytics workflow

The visual analytics workflow of our system is shown in Fig. 3. It contains four main steps, including vectorization of dynamic network, structure extraction via topic models, projection to reveal the evolution, and, finally, visualization and interaction.

3.3.1 Vectorization of dynamic network

We model a dynamic network Γ as a sequence of networks:

$$\Gamma = (G_1, G_2, \dots, G_T) \quad (1)$$

where $G_i = (V_i, E_i)$ is a weighted network at the i^{th} time step with entity set V_i and link (connection) set E_i . Let V be the union of all node sets:

$$V = \cup_{i=1}^T V_i \quad (2)$$

The goal of vectorization is to obtain an input matrix for the LDA model. As shown in Fig. 2, we define G as text corpus in the LDA model, map each G_i to a document, and treat links E_i in the network G_i as its words. In addition, we define the weights of links as the frequencies of words. The weight of a link $w(e_{ij})$ could be defined by its qualitative attributes:

$$w(e_{ij}) = \alpha \times val(e_{ij}) + (1 - \alpha) \times top(e_{ij}), \quad (3)$$

where $val(e_{ij})$ represents the link's attribute values, while $top(e_{ij})$ denotes the importance of its end nodes. In other words, the link's importance depends on its own properties (e.g., relationship strength), as well as the nodes it connects. α is a task-dependent parameter used to balance the two aspects. If $\alpha = 1$, all links will be weighted merely by their properties. If $\alpha = 0$, links of the important nodes will be emphasized. In our cases, we set α to 0.5.

A network represented by $|V| \times |V|$ adjacency matrix is linearized to a $1 \times |V|^2$ row-vector. In the row-vector, each column represents a link, and the value is the link weight. Then we stack these row-vectors of all time steps to form a $T \times |V|^2$ matrix M (Fig. 3). The M is the input of topic models.

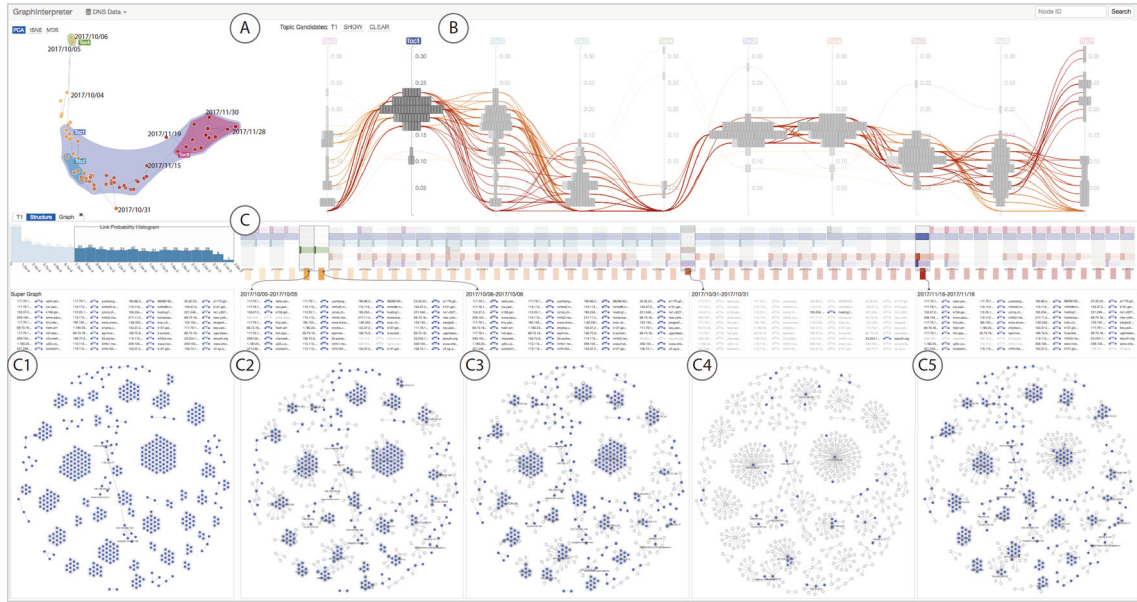


Fig. 1 GraphInterpreter, a novel visual analytics system, consists of three linked views: **a** The projection view reduces networks as points into two-dimensional space; **b** the parallel coordinate view shows the network descriptors; and **c** the small multiples view visualizes the structure in node-link diagram

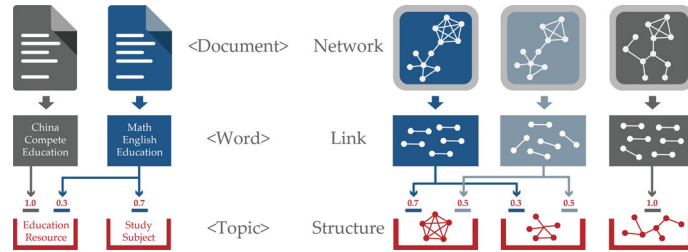


Fig. 2 Concept correspondences of the LDA model in text analysis and our dynamic network analysis. Networks at different time steps are mapped to documents, and links are words. With this input, LDA models could extract a list of structures fused from links and describe the networks by extracted structures

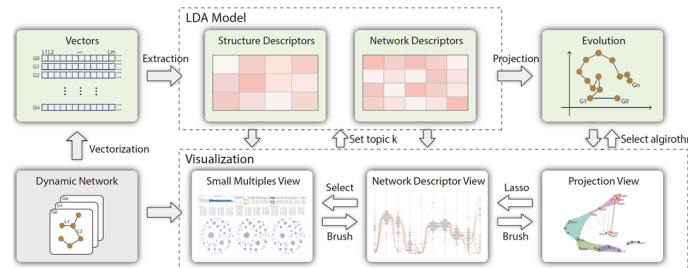


Fig. 3 The visual analytics approach of GraphInterpreter contains four steps: (1) vectorization of dynamic network, (2) structure extraction via topic models, (3) projection to reveal the evolution, and (4) visualization and interaction for exploration and interpretation

3.3.2 Structure extraction via topic models

Here, we employ the LDA model in dynamic network to extract structures that are hard to define in advance. These structures define the similarities between networks and explain in which parts they are similar.

For the matrix M describing a dynamic network, its size is always enormous, especially when the number of time steps are large and the networks are complex, which could cost lots of time in the computation.

However, the matrix is usually very sparse, as there are always large number of nodes which has few connections to other nodes. Moreover, links with higher weights are much more important than those with low values. Based on these considerations, we derive the input matrix M' by removing the columns k in M , if $\sum_{i=1}^T w_{ik}$ is less than the threshold c . After the removal, the column size of M' is much smaller than that of M . As a result, it costs less time in the computation compared to the original matrix.

In text analysis, the LDA model output consists of two distributions: document descriptors and topic descriptors, respectively. A document descriptor is a probability distribution of the document on extracted topics, which describes the feature of the document. A topic descriptor is a probability distribution of the extracted topic on all input words. Typically, topics with high probability in a document descriptor are considered as salient topics in the document, while words in a topic descriptor with high probability are referred to as keywords in the topic. Similarly, in the LDA model input for dynamic networks, two probability distributions in the output are known as network descriptors (network descriptor vectors) and structures descriptors (structures descriptor vectors), respectively. Salient structures in a network and salient links in an extracted structure are used to describe their features. The characteristics of these two kinds of descriptors provide us a novel way to summarize and interpret the evolution of dynamic network.

3.3.3 Projection for network evolution

For a dynamic network, we use network descriptors to describe networks at each time step. Compared with using $1 \times |V|^2$ vectors to describe networks, these descriptors have fewer dimensions. However, it is still not intuitive and hard to comprehend. Hence, we propose to use dimensionality reduction techniques (van der Maaten and Postma 2008) to project these network descriptors into two-dimensional space while preserving their features in high dimensions. Networks with similar network descriptors are close to each other in the projection which could be used to discover stable states, recurrent states, outlier states, and transitions in the evolution of dynamic network.

There are two groups of dimensionality reduction techniques, including linear techniques, such as Principal Component Analysis (PCA) (Pearson 1901), and nonlinear techniques, such as Multidimensional Scaling (MDS) (Kruskal 1964) and t-Distributed Stochastic Neighbor Embedding (t-SNE) (Hinton 2008). van der Maaten and van den Herik (van der Maaten and Postma 2008) take comprehensive comparative review on dimension reduction methods. Specifically, the work investigates the nonlinear dimension reduction methods as opposite to transitional linear ones. They identify the strong performance of nonlinear methods on artificial tasks but not on real-world tasks. However, the selection of dimension reduction method in practice deeply depends on concrete tasks and data characteristics. To facilitate users' exploration, in our system, we support both linear and nonlinear techniques for dimensionality reduction.

4 Visualization and interaction

To facilitate the exploration and interpretation of the dynamic network evolution, we derive three design goals from the analysis tasks introduced in Sect. 3:

- Provide an overview of the dynamic network evolution.
- Show network descriptors based on semantic structures.
- Visualize the semantic structures to support evolution interpretation.

To achieve these design goals, we propose GraphInterpreter, a novel visual analytics system for summarizing and interpreting the evolution of dynamic networks with semantic structures extracted via LDA model. The system contains three linked views (Fig. 3), including projection view, network descriptor view, and small multiples view. Each of these views is tailored to one of the three design goals. In this section, we will introduce the visual design of these views.

4.1 Projection view

The goal of the projection view (Fig. 3a) is to show the evolution of dynamic network. To provide an overview, we project networks at each time step as points in 2D space by reducing their descriptor vectors. Points that are close to each other indicate that the corresponding networks are similar. To enhance the perception of time, sequential points in time are connected by curve paths and colored according to their time steps. Lighter orange indicates earlier time steps, while the current time steps are encoded in darker red.

This color mapping is consistent across all the views. As a result, a cluster with similar colors indicates that networks are quite similar during this period. It is called stable state. If a cluster has quite different colors, it means that networks from different time steps have similar descriptor vectors, indicating a recurrent state.

To help users gain insight of the evolution interpretation, we use convex hulls to highlight points with higher values of the same extracted structures based on a user-defined threshold. In other words, each convex hull represents a structure and contains points that have a high probability of belonging to that structure. The threshold value would influence the number of convex hulls. Lower threshold generates multiple convex hulls. Their overlapping relations show the transitions between evolution states, while higher threshold would generate convex hulls representing salient structures. The color of convex hulls is consistent with the color of extracted structures. As mentioned before, we also use color to encode time, but using different color schemes for time and structures could potentially confuse users during exploration. To address this issue, we allow users to change the color mapping for time steps to grayscale.

To get a suitable overview, users are supported in selecting different dimensionality reduction algorithms based on the features of the data. Additionally, users are allowed to lasso several points and investigate why they are close to or far apart from each other by analyzing the network descriptors in the network descriptor view or select a structure to explore the topology in the small multiples view.

4.2 Network descriptor view

The network descriptor view (Fig. 3b) visualizes network descriptors to facilitate the understanding of similarities or differences between networks. In the view, each axis (dimension) represents an extracted structure, and data items are descriptors.

Generally, in parallel coordinate diagram, using polylines to visualize data items would cause visual clutter. If data items have the same value in some dimensions, some parts of the polylines are overlapped. This phenomenon impedes users perceiving the number of data items with a specific value in a dimension. To alleviate this problem, we discrete the range value of each dimension into several bins. The value interval $Interval_{ij}$ of the j^{th} bin in the i^{th} dimension with N bins in total is defined as

$$Interval_{ij} = [i \times \theta, (i + 1) \times \theta], \theta = C_{max}/N \quad (4)$$

where C_{max} is the maximal value among all descriptor vectors in all dimensions. The value of b_{ij} is the number of descriptor vectors with the value of i^{th} dimension belonging to the interval $Interval_{ij}$. As the number of items increases significantly, it becomes possible to allocate recognizable width to individual rectangle, representing items. The rectangles in the same bin will be merged into a unified bar. And the total width of bar would be compressed and normalized. That enhances scalability, albeit at the cost of individual item identification.

In the network descriptor view, network descriptors in bins are visualized as small rectangles. Besides, curved paths are used to produce more readable visualization (Fig. 4). Suitable curvature direction and strength are set to group neighboring edges and reduce the visual clutter. The color of a curved path is assigned by the time steps of the corresponding network.

Exploration is supported by cross-filtering techniques based on brushing. Cross-filtering enables users to select networks with multiple requirements and investigate their relationships in the projection view. The selected vectors are highlighted in the network descriptor view. Simultaneously, the corresponding network points in projection view would be highlighted with high opacity, and the top k most relevant points would be shown in a little lower opacity. The constant k is set by users, and the relevant score of the i^{th} network to a set of selected network vectors $D = [d_0, d_1 \dots d_n]$ is defined as

$$Relevance_i = \sum_{j=0}^n \frac{1}{||d_i - d_j|| + \epsilon}, d_j \in D, \quad (5)$$

where d is the descriptor vector of the network and ϵ is a small constant to avoid zero as denominator. To inspect the topology of a single structure or multiple structures simultaneously, users are supported to explore the topology in small multiples view.

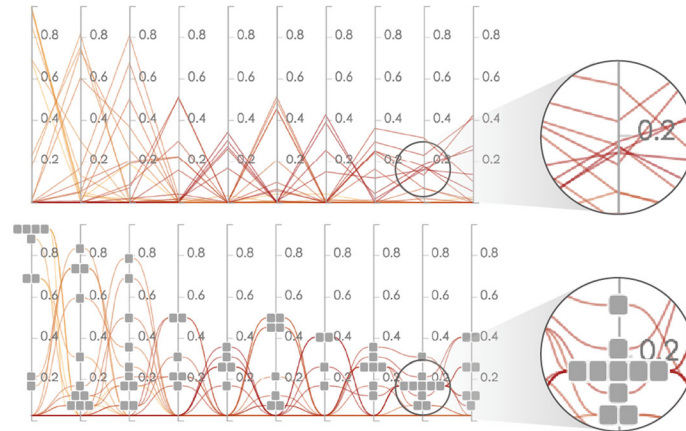


Fig. 4 The comparison between the parallel coordinate diagram in polylines and network descriptor view with curve path and discretization

4.3 Small multiples view

This view uses small multiples to reveal the topology of extracted structures and their evolution (Fig. 3c). It consists of three main components, including a histogram, a timeline, and small multiples.

The histogram part shows the structure descriptors intuitively and facilitates the selection of salient links. The histogram in this view encodes the probabilities on the x-axis, and the height of each bar indicates the number of links belonging to the corresponding value interval. The shape of the histogram reveals the features of the selected structures. If there are several bars on the right, it indicates that these salient links describe the structures well. Users can select these links for further exploration by brushing.

The goal of the timeline is to provide an overview of how extracted structures behave over time. In this part, we encode the salient structures as small bars, with each bar colored according to the structure it represents. The horizontal position of each bar encodes the time it belongs to, and the vertical position is determined by the structure it represents. To enable users to easily relate bars with the same color, we add a dashed line to connect bars that are not adjacent.

The goal of the small multiples is to show the topology of selected structures, including extracted structures and combined structures. When users select multiple extracted structures to explore simultaneously, the system generates a combined structure. In the combined structure, the probability of each link is defined as the maximal value of the link in the selected structures. The combined structures are particularly important when networks are complex and described by multiple structures. However, we do not distinguish between these two kinds of structures, as the related visualization and interaction are the same.

In the small multiples, salient links selected in the histogram are shown in node-link diagram. To reduce the visual clutter, cliques and clusters in star are detected and visualized in hexagonal honeycomb layout. The algorithm for arranging cliques into hexagonal honeycomb layout consists of three main steps (Fig. 5):

- Divide the set of nodes V in the clique into two sets V_{inside} and $V_{outside}$ based on whether they connect nodes outside the clique or not.
- Calculate a hexagonal honeycomb layout based on the number of nodes in the clique.
- Assign nodes in V to hexagons in the honeycomb layout. Nodes in V_{inside} are assigned inside, while nodes in $V_{outside}$ are placed in the outside layer of the honeycomb layout.

Similarly, for a cluster in star style, we position leaf nodes around the central hub, located at the center of the honeycomb layout. To emphasize bridge nodes that link the clique to the outside, their hexagonal backgrounds are colored in gray. The top k salient links are displayed at the top of node-link layout to summarize the structure descriptors. We represent the probabilities as the number of dots.

After selecting a set of interesting links in the histogram, the timeline uses highlighted bars to show how many selected links are also important in the corresponding structures. Here, importance is a relative concept. If the probability of a selected link in a structure is higher than that in histogram, we consider that this link is important in this structure. Meanwhile, a small bar is shown at each time step to reveal how many selected links have appeared in the network. These bars are colored according to the time index. With this design, users are supported in gaining a basic understanding of how the structures behave over time. To

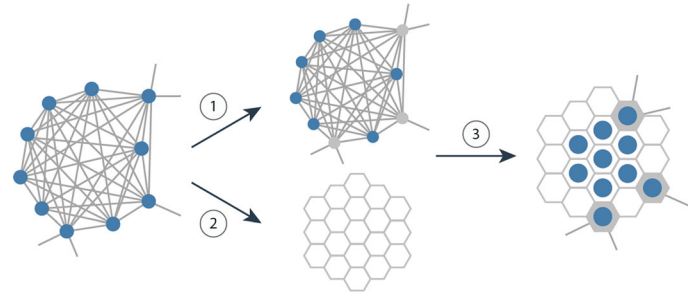


Fig. 5 The compact layout algorithm based on honeycomb for a clique contains three main steps: (1) divide the nodes into two sets; (2) calculate a basic hexagonal honeycomb layout, and (3) assign the nodes in the honeycomb

Before t_{start}	During $[t_{start}, t_{end}]$		After t_{end}	Before t_{start} && After t_{end}
	Appear at $t_{start} - 1$	Disappear at $t_{start} - 1$		

Fig. 6 There are five kinds of styles for nodes based on their appearing time. For a time range, the appearing nodes are visualized as solid circles, while other nodes are hollow circles with dotted stroke

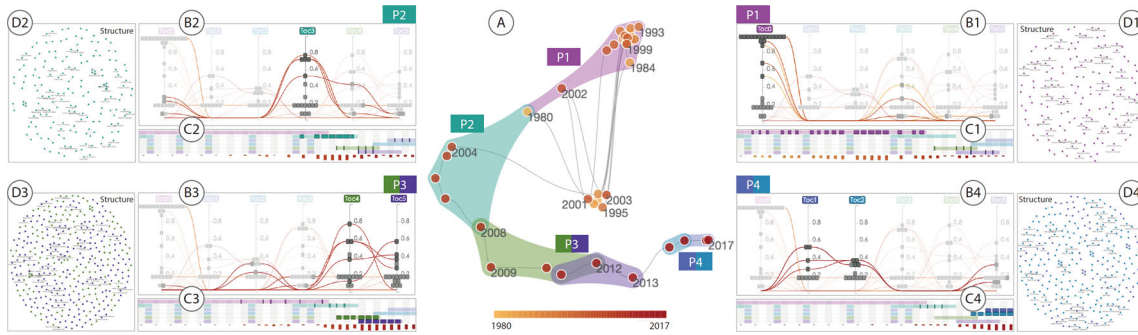


Fig. 7 The visual exploration and interpretation of evolution patterns in the collaboration network derived from publications in AAAI conference. The projection view **a** reveals four evolution states (P1–P4), which show the different stages during the development of the conference

facilitate further exploration, brushing is supported to select a time range. The small multiples show how the structures behave during the selected time range. To highlight how the selected structures evolve, nodes are divided into four types and are encoded in different styles (Fig. 6). For the dotted circles, the size of dots is the same while their intervals are mapped to the time difference. Besides, multiple brushes are supported for comparison.

5 Case studies

In this section, we present the visual analysis of three real-world datasets, to demonstrate the effectiveness of our approach for analyzing different types of dynamic networks.

5.1 Case 1: Dynamic collaboration network

Here, we would like to explore dynamic collaboration networks in two fields, artificial intelligence (AI) field and visualization (VIS) field, respectively. During the exploration, we will focus on analyzing the similarities and differences in their evolutions.

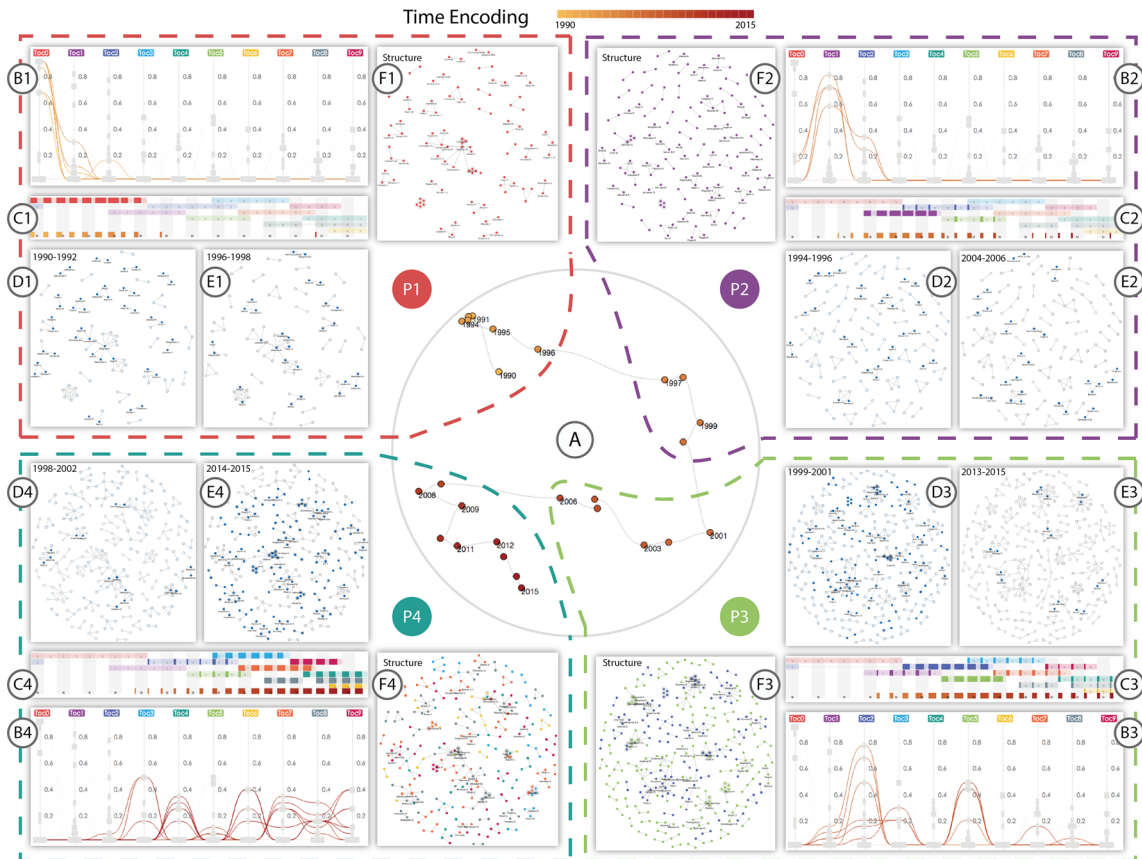


Fig. 8 The visual exploration and interpretation of evolution patterns in the collaboration network derived from publications in the IEEE VIS conference. The projection view **a** reveals four evolution states (P1–P4), which show the different stages during the development of the conference

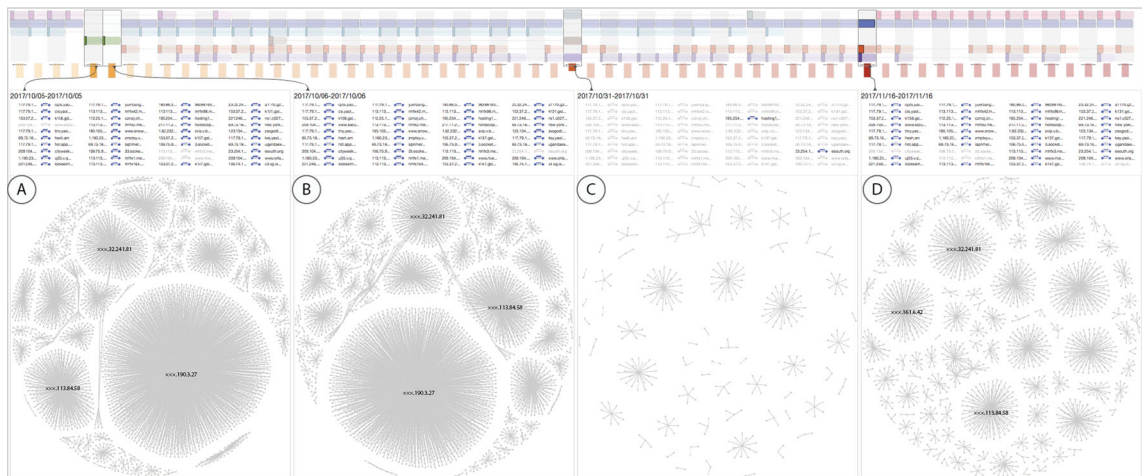


Fig. 9 From left to right, there are original networks in October 5, October 6, October 31, and November 16. Compared with the network in a common day, networks in October 5 and October 6 have a large cluster, while the network in October 31 is quite small

5.1.1 Collaboration network in AAAI conference

In this case, we investigate the evolution of dynamic collaboration networks in the Association for the Advancement of Artificial Intelligence (AAAI) conference. The original data are collected from DBLP database, which contains papers published from 1980 to 2017. During this 38-year period, there were six years without any publication, namely 1981, 1985, 1989, 1995, 2001, and 2003, which correspond to the gap years when the conference is not held. We extracted author information from the data to construct the collaboration network for each year. Entities in the network represent authors, while links represent the collaborated publications. After the construction, we obtained a total of 16,500 entities, 36,871 links, and 38 time steps.

In the analysis, our main focus is on the general evolution of academic collaborations, which can reflect the development of AI field over the past few decades. During the preprocessing stage, we set the number of topics to 6 in the LDA model. The number of topics was chosen after discussions with domain experts, and several attempts were made to select the number that worked best. After comparing effects of different dimension reduction methods, the MDS is chosen to generate the projection view (Fig. 7a), with contours to indicate different evolution structures. From the projection, we observed that the evolution is relatively smooth, as there are always transition points between neighboring contours. We also identified four main states (i.e., P1, P2, P3, and P4), as well as a central cluster that does not possess any featured structure.

A closer examination of the central cluster reveals that it corresponds precisely to the 6 gap years during which no publication data were recorded. This allows users to promptly identify the missing data in the projection, highlighting the capability of our method to handle complex data with diverse patterns and exceptions.

Details of the four main states are shown in Fig. 7. For each state, we use three views to illustrate its features. The network descriptor view (Fig. 7b) shows the weights of extracted structures. Topologies of the dominant structures are shown in the node-link diagram (Fig. 7d). The timeline view displays saliency of each state across different time steps (Fig. 7c).

State P1 describes the collaborations during the early stage of the conference, from 1979 to 2002. The closely distributed points (Fig. 7a) indicate that the collaboration networks are relatively stable during this period. The network descriptor view (Fig. 7b1) shows that the first structure with high weights can summarize the networks quite well. Then we select salient links in this dominant structure to study its topology. The resulting graph is small and sparse (Fig. 7d1). This indicates that the community was small with simple collaborations in the early years of AAAI. In the timeline view (Fig. 7c1), the highlighted bars denote how many selected links are salient across different years. From this view, we can see that most active authors in the early years are not active now, as the highlighted bars become much smaller after 2006.

Similarly, we find that points of states P2 and P3 (from 2003 to 2013) are relatively far apart in the projection. It indicates a rapid development of the collaboration network with more and more authors joining the community. In contrast, P4 represents a stable state as the community begins to take shape.

Comparing Fig. 7b1–b4, we also see that the highlighted paths become smoother and smoother. In other words, the collaboration networks are becoming more and more complex. They can no longer be summarized by a certain kind of dominant structure. On the other hand, the featured structures (Fig. 7d1–d4) are becoming increasingly complicated with more entities and links. This confirms the previous finding, as more complex structures are required to describe more complex networks.

From the above analysis, we have discovered different stages of the AAAI conference. In the early years (from 1979 to 2002), collaborations were relatively simple and stable. In the past decade, the collaboration networks became more and more complex as many new researchers joined the community. In more recent years (from 2013 to now), the collaborations have become more stable as the community has begun to take shape. The development of academic collaborations indicates that the AI field has been developing for many years and is still active and growing.

5.1.2 Collaboration network in VIS conference

In this case, we analyze the evolution of the collaborations in the IEEE VIS (previously named VisWeek) conference. The dynamic collaboration network is extracted from papers published in the IEEE VIS conference from 1990 to 2015 (Isenberg et al. 2015). There are, in total, 4,813 entities and 14,033 links. Here, we focus on identifying and interpreting the evolution patterns of the collaboration networks, which could reflect the development process of the VIS field from the perspective of researchers' cooperations.

In the preprocessing step, we discuss with domain experts and set the topic number to 10 in LDA model. Then, we apply MDS to the network descriptor vectors to project networks as points into two-dimensional space. The visual exploration result is shown in Fig. 8.

First, from the projection view (Fig. 8a), we can observe that the evolution of the dynamic collaboration network consists of four main states (Fig. 8P1–P4). For each state, we use a snapshot from the parallel coordinate view to show main extracted structures used to describe the corresponding networks (Fig. 8b). To explore the topology of these main structures, we select the relatively salient links and use the timeline view to provide an overview of how these selected links behave at different time steps (Fig. 8c). We also include three small multiples to show the topology of the structures (Fig. 8f) and their changes at the beginning and at the end of each state, respectively (Fig. 8d, e).

The cluster P1, which covers the period from 1990 to 1996, describes collaborations during the early stage of the development of the conference. The points in P1 of Fig. 8a that are near to each other indicate that the collaboration networks are relatively stable. The parallel coordinate view (Fig. 8b1) highlights the corresponding network descriptor vectors. We find that most vectors have very high probability (≥ 0.6) in the same structure, indicating that this structure could describe these networks quite well. Then, we select salient links in this structure to explore its topology, which is shown in Fig. 8f1. The graph is relatively small and sparse, suggesting that the number of salient authors during this early stage is quite small. The timeline view (Fig. 8c1) uses the width of highlighted bars to represent how many selected links are also salient at different years. From this view, we could infer that most active authors during the early stage are no longer active, as there are only two small highlighted bars from 2000 to the present.

With this analytics method, we discovered that state P2 and state P3 describe the state transitions during the development period of the conference from 1997 to 2006, while state P4 represents the collaboration during the stable development state. The projection view (Fig. 8a) shows that points in states P1 and P4 are quite near to each other, which means that the collaboration networks during these two periods are quite stable, while points in states P2 and P3 are quite far apart, which indicates that the collaboration networks changed a lot and the research field is developing rapidly with more and more authors joining in. From Fig. 8b1–b4, we can see that the highlighted paths have more and more peaks, indicating that the corresponding collaboration networks became more and more complex and required more structures for description. The small multiples (Fig. 8f1–f4) became more and more dense with more entities and links, supporting our findings that the collaboration networks became more complex as the conference develops. Meanwhile, we find that the highlighted bars in Fig. 8c3, c4 would cover a longer time range than those in Fig. 8c1, c2. It means that more and more authors have focused on this research field for a long time and play important roles. After checking the history of the IEEE VIS conference, we find that the conference changed the name in 1995 and first held the VAST conference in 2006. This fact is in good agreement with the evolution of collaboration networks that we have discovered.

After exploration, we find that the collaborations in the AAAI conference and the IEEE VIS conference have the same evolution patterns. Early in the conference, the collaborations are quite stable, and the size of corresponding networks is relatively small. And then, the networks become larger and more complicated with the development of the field. From the exploration, we can conclude that both AI and VIS field are hot research area. Both of them have large communities and attract more and more researchers to take part in.

5.2 Case 2: DNS parsing log analysis

In this case, we use our system to analyze the evolution of IP-DNS connection networks. The data analyzed consisted of DNS parsing logs collected by domain experts between October 1 to November 30, 2017. We derived the IP-DNS networks with a time granularity of one day. In the network, each node represents an IP address or a domain name. If an IP address can be parsed to a certain domain name, a link between the two nodes was created. The dynamic network contains a total of 26,714 nodes, 26,762 links, and 61 time steps.

We generate 10 topics from the LDA model. And we select PCA algorithm after evaluation in the projection view (Fig. 1b). From this view, we observe that most points are close to each other, indicating that the IP-DNS connection networks are relatively stable. Specifically, the evolution of the network contains two stable states and two outlier states. As shown in the network descriptor view (Fig. 1b), most network descriptors have low probabilities on the extracted structures, indicating that the networks are difficult to describe using any individual structure.

The projection view (Fig. 1a) shows two large convex hulls colored purple and red, respectively. We can also identify three outliers in the projection: October 5, October 6, and October 31.

We first explore the purple convex hull, whose large size suggests that this state may describe the common topology of daily IP-DNS connection networks. In the timeline part of the small multiples view (Fig. 1c), the bars highlighted in purple appear on all days except three days: October 5, October 6, and October 31. These are exactly the outliers we have discovered in the projection view.

The small multiples (Fig. 1c2, c3) show that almost all salient links have also appeared on October 5 and October 6. However, the dominant structure on October 5 and October 6 is an outlier structure in green (Fig. 1b), which is quite strange. In the hexagonal honeycomb layout, we can see the state of the entire network more clearly thanks to our concise presentation of cliques and clusters. However, if we want to look specifically at the details of a particular topology, we can switch to the origin layout in small multiples. We further explore the original networks (Fig. 9) with November 16 being a reference state. We find that the networks on October 5 and October 6 (Fig. 9a, b) also contain the same topology as in November 16 (Fig. 9d). However, they also carry a large radial cluster where a virtual IP is connected to many domain names for different websites. Domain experts describe this cluster as an unexpected finding, which is often hard to detect in their experience. This structure is probably a defense behavior, as explained by the domain experts. Some websites will use a virtual IP to transfer heavy network flows caused by DDOS attacks. In other words, the green convex hull describes the defensive network state.

Another large convex hull in red (Fig. 1b) describes a network state during two weeks from November 17 to November 30. Its featured structure is very similar to the outlier state during October 5 and October 6, with a major subgraph connecting a virtual IP to many domain names. They seem to have been attacked during the week and used the same mechanism for defense. Different from the previous case, all domain names here have the same suffix. We infer that these are websites of the same organization.

The outlier on October 31 is also worth studying. It seems that the IP-DNS connection network is too simple compared to the network in a stable state (Fig. 9c, d). Domain experts explain that the data size is exceptionally small on that day due to some failures in the data collection process.

6 Discussion

In this paper, we propose a novel visual analytics approach based on LDA model to extract semantic structures for dynamic network evolution exploration and interpretation. We demonstrate the effectiveness of our approach and system on two different datasets. In addition to these two types of data, other dynamic network data such as social networks, financial networks, and other networks can also be explored and interpreted using our system. There are many parameters that need to be set and multiple alternative algorithms that could be chosen at each step of our proposed method. The process of setting parameters and selecting related algorithms is dependent on the characteristics of the dynamic network being analyzed. For instance, the input parameters of the LDA model, such as the number of topics, are used to specify the model characteristics, which are highly dependent on the features of the data being analyzed. Therefore, it might be beneficial for users to collaborate with domain experts.

For the visualization part, there are two scalability concerns, namely, the number of time steps and the size of the network. If the number of time steps is too small, the proposed method may be less effective, as it can be difficult for the LDA model to extract meaningful structures to cluster networks based on their similarities. Therefore, when using this dynamic network analytics approach, it is better to analyze networks that evolve over more than 20 time steps. As for the network size at each time step, it can affect the understanding of the evolution patterns. To handle large networks, we propose two main solutions: Firstly, our system supports users in selecting a suitable number of salient relationships for further exploration through brushing. Secondly, we propose a novel node-link diagram with a compact honeycomb layout for cliques, which can handle dense and large networks.

7 Conclusions and future work

In this work, we introduce a novel approach for analyzing the evolution of dynamic networks based on the LDA model. We apply the LDA model to dynamic networks by defining *networks* at different time steps as *documents* and *links* in networks as *words*. With this definition, the LDA model can extract a list of structures fused from links and describe the networks by extracted structures with probabilities. We also provide a novel visual analytics system, GraphInterpreter, to facilitate the exploration and interpretation of

dynamic network evolution with extracted structures. With two case studies of real-world datasets, we have demonstrated the effectiveness of our approach and the usefulness of our visual analytics system.

In future research, we would like to improve our work from the following perspectives: First, we would like to update the mapping method when employing the LDA model on dynamic networks. On one hand, we would consider nodes' attributes when calculating the weight of links, which is the input of the LDA model. On the other hand, to gain semantic structures, it is useful to consider more topological information in our method, such as betweenness and centrality. Second, we would like to support comparison between structures in our visual analytics system, to help users gain an insight of the similarities and differences among extracted structures.

Acknowledgments This work is supported by NSFC, No. 62272012.

References

- Abello J, Hadlak S, Schumann H, Schulz H-J (2014) A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Trans Visual Comput Graphics* 20(3):337–350. <https://doi.org/10.1109/tvcg.2013.109>
- Bach B, Pietriga E, Fekete J-D (2014) Graphdiaries: animated transitions and temporal navigation for dynamic networks. *IEEE Trans Visual Comput Graphics* 20(5):740–754. <https://doi.org/10.1109/tvcg.2013.254>
- Beck F, Burch M, Diehl S, Weiskopf D (2014) The state of the art in visualizing dynamic graphs. *EuroVis STAR*. <https://doi.org/10.2312/eurovisstar.20141174>
- Beck F, Burch M, Diehl S, Weiskopf D (2016) A taxonomy and survey of dynamic graph visualization. In: *Computer graphics forum*, vol 36, pp 133–159. Wiley Online Library, 2016. <https://doi.org/10.1111/cgf.12791>
- Blei DM, Ng AY, Jordan MI (2002) Latent Dirichlet allocation. *J Mach Learn Res* 3:2003
- Burch M, Fritz M, Beck F, Diehl S (2010) Timespidertrees: a novel visual metaphor for dynamic compound graphs. In: 2010 IEEE symposium on visual languages and human-centric computing, pp 168–175. IEEE. <https://doi.org/10.1109/vlhcc.2010.31>
- Cha Y, Cho J (2012) Social-network analysis using topic models. In: *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval*, pp 565–574. ACM. <https://doi.org/10.1145/2348283.2348360>
- Che L, Liang J, Yuan X, Shen J, Xu J, Li Y (2015) Laplacian-based dynamic graph visualization. In: 2015 IEEE pacific visualization symposium (PacificVis), pp 69–73. IEEE. <https://doi.org/10.1109/pacificvis.2015.7156358>
- Dang TN, Pendar N, Forbes AG (2016) Timearcs: visualizing fluctuations in dynamic networks. In: *Computer graphics forum*, vol 35, pp 61–69. Wiley Online Library. <https://doi.org/10.1111/cgf.12882>
- Diehl S, Görg C, Kerren A (2001) Preserving the mental map using foresighted layout. In: *Data visualization 2001*, pp 175–184. Springer. https://doi.org/10.1007/978-3-7091-6215-6_19
- Diehl S, Görg C (2002) Graphs, they are changing- dynamic graph drawing for a sequence of graphs. In: *Proceedings of 10th international symposium graph drawing (GD 2002)*, number 2528 in *Lecture Notes in Computer Science, LNCS*, pp 23–31. Springer. [10.1007/3-540-36151-0_3](https://doi.org/10.1007/3-540-36151-0_3)
- Dwyer T, Eades P (2002) Visualising a fund manager flow graph with columns and worms. In: *Information Visualisation, 2002. Proceedings. Sixth international conference on*, pp 147–152. IEEE. <https://doi.org/10.1109/iv.2002.1028770>
- Erten C, Harding PJ, Kobourov SG, Wampler K, Yee G (2003) Graphael: graph animations with evolving layouts. In: *International symposium on graph drawing*, pp. 98–110. Springer. [10.1007/978-3-540-24595-7_9](https://doi.org/10.1007/978-3-540-24595-7_9)
- Erten C, Kobourov SG, Le V, Navabi A (2003) Simultaneous graph drawing: layout algorithms and visualization schemes. In: *International symposium on graph drawing*, pp 437–449. Springer. <https://doi.org/10.7155/jgaa.00104>
- Falkowski T, Bartelheimer J, Spiliopoulou M (2006) Mining and visualizing the evolution of subgroups in social networks. In: *Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence*, pp 52–58. IEEE Computer Society. <https://doi.org/10.1109/wi.2006.118>
- Forrester D, Kobourov SG, Navabi A, Wampler K, Yee GV (2004) Graphael: a system for generalized force-directed layouts. In: *International symposium on graph drawing*. Springer. [10.1007/978-3-540-24595-7_9](https://doi.org/10.1007/978-3-540-24595-7_9), pp 454–464
- Gorochowski TE, di Bernardo M, Grierson CS (2012) Using aging to visually uncover evolutionary processes on networks. *IEEE Trans Visual Comput Graphics* 18(8):1343–1352. <https://doi.org/10.1109/tvcg.2011.142>
- Greilich M, Burch M, Diehl S (2009) Visualizing the evolution of compound digraphs with timearctrees. In: *Computer graphics forum*, vol 28, pp 975–982. Wiley Online Library. <https://doi.org/10.1111/j.1467-8659.2009.01451.x>
- Hadlak S, Schumann H, Cap CH, Wollenberg T (2013) Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Trans Visual Comput Graphics* 19(12):2267–2276. <https://doi.org/10.1109/tvcg.2013.198>
- Henderson K, Eliassi-Rad T (2009) Applying latent dirichlet allocation to group discovery in large graphs. In: *Proceedings of the 2009 ACM symposium on applied computing*, pp 1456–1461. ACM. <https://doi.org/10.1145/1529282.1529607>
- Hinton GE (2008) Visualizing high-dimensional data using t-sne. *Vigiliae Christianae* 9(2):2579–2605
- Huang ML, Eades P, Wang J (1998) On-line animated visualization of huge graphs using a modified spring algorithm. *J Visual Lang Comput* 9(6):623–645. <https://doi.org/10.1006/jvlc.1998.0094>
- Isenberg P, Heimerl F, Koch S, Isenberg T, Xu P, Stolper C, Sedlmair M, Chen J, Möller T, Stasko J (2015) Visualization publication dataset. dataset: <http://vispubdata.org/http://vispubdata.org/~>
- Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1):1–27. <https://doi.org/10.1007/bf02289565>

- Pearson K (1901) On lines and planes of closest fit to systems of points in space. *Phil Mag* 2(6):559–572. <https://doi.org/10.1080/14786440109462720>
- Rufiange S, McGuffin MJ (2013) Diffani: visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Trans Visual Comput Graphics* 19(12):2556–2565. <https://doi.org/10.1109/tvcg.2013.149>
- Shi L, Wang C, Wen Z (2011) Dynamic network visualization in 1.5 d. In: 2011 IEEE Pacific visualization symposium, pp 179–186. IEEE. <https://doi.org/10.1109/pacificvis.2011.5742388>
- Steiger, M, Bernard J, Mittelstädt S, Lücke-Tieke H, Keim D, May T, Kohlhammer J (2014) Visual analysis of time-series similarities for anomaly detection in sensor networks. In: *Computer graphics forum*, vol 33, pp 401–410. Wiley Online Library. <https://doi.org/10.1111/cgf.12396>
- van den Elzen S, Holten D, Blaas J, van Wijk JJ (2016) Reducing snapshots to points: a visual analytics approach to dynamic network exploration. *IEEE Trans Visual Comput Graphics* 22(1):1–10. <https://doi.org/10.1109/tvcg.2015.2468078>
- van der Maaten L, Postma EO (2008) and H. A comparative review, J. van den Herik. Dimensionality reduction
- Vehlow C, Beck F, Auwärter P, Weiskopf D (2015) Visualizing the evolution of communities in dynamic graphs. In: *Computer graphics forum*, vol 34, pp 277–288. Wiley Online Library, 2015. <https://doi.org/10.1111/cgf.12512>
- Vehlow C, Beck F, Weiskopf D (2016) Visualizing dynamic hierarchies in graph sequences. *IEEE Trans Visual Comput Graphics* 22(10):2343–2357. <https://doi.org/10.1109/TVCG.2015.2507595>
- von Landesberger T, Diel S, Bremm S, Fellner DW (2015) Visual analysis of contagion in networks. *Inf Vis* 14(2):93–110. <https://doi.org/10.1177/1473871613487087>
- Wang Y, Wang Y, Sun Y, Zhu L, Lu K, Fu C-W, Sedlmair M, Deussen O, Chen B (2017) Revisiting stress majorization as a unified framework for interactive constrained graph visualization. *IEEE Trans Visual Comput Graph.* <https://doi.org/10.1109/tvcg.2017.2745919>
- Zhang H, Qiu B, Giles CL, Foley HC, Yen J (2007) An lda-based community structure discovery approach for large-scale social networks. In: *Intelligence and security informatics, 2007 IEEE*, pp 200–207. IEEE. <https://doi.org/10.1109/isi.2007.379553>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.