Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

RCAnalyzer: visual analytics of rare categories in dynamic networks*

Jia-cheng PAN†[1,2], Dong-ming HAN[1,2], Fang-zhou GUO[1], Da-wei ZHOU[3], Nan CAO[4], Jing-rui HE[3], Ming-liang XU[5,6], Wei CHEN†‡[1,2]

[1]*State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310058, China*
[2]*Zhejiang Lab, Hangzhou 311100, China*
[3]*Department of Computer Science and Engineering, Arizona State University, Tempe 85281, USA*
[4]*Intelligent Big Data Visualisation Lab, Tongji University, Shanghai 200082, China*
[5]*School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China*
[6]*Henan Institute of Advanced Technology, Zhengzhou University, Zhengzhou 450001, China*
†E-mail: panjiacheng@zju.edu.cn; chenvis@zju.edu.cn

Received June 24, 2019; Revision accepted Nov. 26, 2019; Crosschecked Jan. 30, 2020

Abstract: A dynamic network refers to a graph structure whose nodes and/or links dynamically change over time. Existing visualization and analysis techniques focus mainly on summarizing and revealing the primary evolution patterns of the network structure. Little work focuses on detecting anomalous changing patterns in the dynamic network, the rare occurrence of which could damage the development of the entire structure. In this study, we introduce the first visual analysis system RCAnalyzer designed for detecting rare changes of sub-structures in a dynamic network. The proposed system employs a rare category detection algorithm to identify anomalous changing structures and visualize them in the context to help oracles examine the analysis results and label the data. In particular, a novel visualization is introduced, which represents the snapshots of a dynamic network in a series of connected triangular matrices. Hierarchical clustering and optimal tree cut are performed on each matrix to illustrate the detected rare change of nodes and links in the context of their surrounding structures. We evaluate our technique via a case study and a user study. The evaluation results verify the effectiveness of our system.

Key words: Rare category detection; Dynamic network; Visual analytics
https://doi.org/10.1631/FITEE.1900310 **CLC number:** TP311

1 Introduction

In many cases, relations among objects can be modeled as time-evolving networks, such as collaborations among researchers, transactions among traders, and communications in social networks. These relations reflect how individuals act in a network over time and the goals of their activities (Jovanovic et al., 2015). Most individuals in a network behave normally, while a minority may act differently from the others, indicating anomalous situations. Anomalies could be positive, such as superstars in a collaboration network and recipients or benefactors in a financial network, or negative, which may damage the development of the entire graph, such as frauds in a trading network and criminals or spies in a communication network. In either case, finding these anomalous changing behaviors of network structures is valuable.

Most of the existing anomaly detection

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. U1866602, 61772456, U1736109, and 61972122)
ORCID: Jia-cheng PAN, https://orcid.org/0000-0002-8676-9990; Wei CHEN, https://orcid.org/0000-0002-8365-4741

algorithms are automatic, and do not take human insights into account. In contrast, active learning is a special case of machine learning that improves automatic algorithms' performance with human knowledge. Following an active learning procedure, many rare category detection (RCD) methods are thus developed (Pelleg and Moore, 2005; He and Carbonell, 2008, 2009; Huang et al., 2011, 2013), and candidates that are most likely to represent rare categories are detected and labeled by users. The RCD methods are one set of anomaly detection algorithms which recognize abnormal individuals as rare categories because their number is usually very small. Once labeled, the algorithm will propagate the label to the nearby instances which are similar to the labeled one in a feature space. Those representative candidates are usually centers of rare categories. This procedure has one major limitation; i.e., it is still difficult for users to make a correct judgment (regarding whether or not a candidate represents a rare category) given one single data instance with the entire context information missing. This is particularly difficult for detecting rare categories from a dynamic graph as both the temporal information and structural information need to be considered while labeling a candidate. Therefore, visualization could be helpful in terms of supporting interactive data exploration and providing a rich context representation.

However, challenges exist in designing such a visualization system to support the process of RCD in a dynamic network. First, although capturing the temporal dynamics of a changing structure is a problem that has been extensively studied (Beck et al., 2014), none of the existing techniques is developed to support the visualization of rare categories. Second, capturing the changing structures of rare categories in the context of a big dynamic graph is challenging as the rare categories are usually very small and their evolutions could be very likely to be ignored. Third, to better support the decision-making process, the visualization should be able to differentiate different structures in detail, which is not easy to achieve.

To address the above challenges, in this study, we propose a novel visualization system called RCAnalyzer. RCAnalyzer represents a large dynamic network in the form of a series of connected triangular matrices where each matrix represents a snapshot (Fig. 1). Compared to a square matrix, triangles

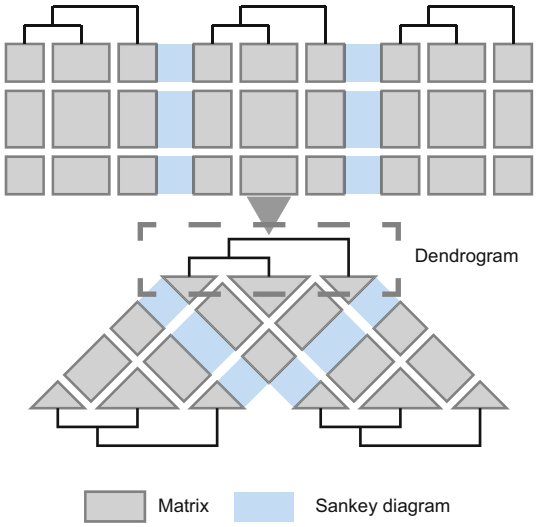

Fig. 1 Basic design of the matrices view, including matrix, Sankey diagram, and dendrogram

are more space-efficient. A hierarchical clustering algorithm and a tree cut algorithm are developed to produce an adaptive "focus+context" view that aggregates the graph structure into a hierarchy, so that a large graph can be fully displayed while showing the detailed structures of potential rare categories. The proposed matrix based visualization facilitates an in-context visual comparison of substructures in a dynamic graph, which improves the efficiency of rare category detection. In particular, this study has the following contributions: (1) a novel tree cut algorithm that produces a multi-focus view to illustrate the substructure details of multiple rare categories in the context of a big dynamic graph, (2) a novel dynamic network visualization design in the form of a series of connected triangular matrices that highlights the detected rare categories in both the temporal and topological context and facilitates the substructure comparison, and (3) an integrated visual analysis system that supports the detection of rare categories and facilitates rare category labeling.

2 Related work

2.1 Dynamic network anomaly detection

Anomaly detection in dynamic networks refers to the detection of anomalous nodes, edges, subgraphs, and time-evolving changes. Several existing surveys have reviewed the most popular anomaly detection methods used in dynamic networks (Bhuyan et al., 2014; Ranshous et al., 2015).

Ranshous et al. (2015) categorized the existing methods into five types: community-based, compression-based, decomposition-based, distance-based, and probabilistic model based methods. For example, in compression-based methods, a graph stream can be divided into multiple segmentations using the minimum description length principle. Anomaly changes can then be detected at the time points when a new segment begins (Sun et al., 2007). Probabilistic model based methods usually construct a "normal" model and use it to detect anomalies which deviate from the "normal" model. For example, when the number of communications deviates from the expected number generated by conjugate Bayesian models, the time point would be considered as an anomaly (Heard et al., 2010).

As we mentioned in Section 1, these anomaly detection studies do not capture user's intention. In contrast, RCD refers to a series of active learning methods which incorporate human knowledge. Many RCD methods require prior information to detect the minority classes (Pelleg and Moore, 2005; He and Carbonell, 2008; He et al., 2008, 2010; Zhou et al., 2015a, 2015b, 2017). However, many data sets do not have any prior information. To avoid this limitation, Huang et al. (2011, 2013) and He and Carbonell (2009) presented a series of prior-free methods. Compactness assumption based methods (He and Carbonell, 2008; He et al., 2008; Zhou et al., 2015b, 2017) assume that the distribution of the major categories is smooth and compact, and compactness isolation assumption based methods (Vatturi and Wong, 2008; Huang et al., 2013) require the rare categories be isolated from the major category. Lin et al. (2018) presented RCLens, a visual analytics system supporting user-guided rare category exploration and identification. RCLens supports users in identifying rare categories in the high dimensional datasets. However, it is not designed for rare category identification in dynamic networks.

2.2 Visualization of anomaly

Many visualization techniques have been developed to aid the detection and analysis of anomalies (Chandola et al., 2009; Haberkorn et al., 2014; Liu et al., 2017; Zhang et al., 2017). Dimension reduction methods, such as principal component analysis (Jolliffe, 1986), and multidimensional visualization techniques, such as parallel coordinate plots (Inselberg, 2009) and DICON (Cao et al., 2011), are commonly used to visualize the data distribution and show outliers with abnormal distribution. In ViDX (Xu et al., 2017), an extended Marey graph was used to show outliers in the manufacturing procedure. Anomalies in network traffic data (Teoh et al., 2002; Tsai et al., 2009; Corchado and Herrero, 2011) and social media data (Thom et al., 2012; Zhao et al., 2014; Cao et al., 2016) have also drawn a lot of attention. Fluxflow (Zhao et al., 2014) detects the diffusion of anomalous information in social media and TargetVue (Cao et al., 2016) uses glyph-based designs to show the anomalous behaviors in online communication systems based on an unsupervised learning model. Wang et al. (2013) presented SentiView to visualize the sentiment in Internet topics and enabled analysts to monitor abnormal events on the Internet. Fan et al. (2019) presented an interactive visual analytics approach, which combines active learning and visual interaction, for detecting anomalies.

Compared to the existing methods, our method focuses on detecting the rare categories in dynamic networks based on RCDs. To the best of our knowledge, there is no existing visualization system that supports users in analyzing and labeling anomalies based on RCDs. Moreover, we have developed a series of interactions which enable users to compare rare categories within the entire dynamic networks.

2.3 Visualization of dynamic networks

There have been a lot of studies on visualization of dynamic networks over the years. Beck et al. (2014) reported the state of the art of dynamic network visualization in a fine survey, who classified the visualization techniques of dynamic networks into animated diagrams (Yee et al., 2001; Bach et al., 2014a) and timelines of a series of static charts, such as node-link diagrams or adjacency matrices. Timelines with matrix- and flow-based representation methods are most relevant to our work. Archambault et al. (2011) found that small-multiple-based techniques have better performance than animation-based techniques.

Matrix-based techniques can be classified into two categories. The first category embeds a timeline into each cell of the matrix. Gestaltlines (Brandes and Nick, 2011), fingerprint glyphs (Oelke et al., 2013), and the horizon graph (Burch et al., 2013) are used to show the evolution of dyadic relations in

a matrix. However, this category of methods does not fit well with large datasets. The second category lays a sequence of adjacency matrices in a certain order (Bach et al., 2014b, 2015; Zhao et al., 2015). van den Elzen et al. (2016) reduced the matrices into points and lay the points using production methods. Both NodeTrix (Henry et al., 2007) and Dendrogramix (Blanch et al., 2015) visualized a static graph by combining several visualization representations. However, they are not designed for visualizing dynamic networks, and cannot show the change of networks properly.

Flow-based techniques use flow metaphors to represent the evolution of communities in networks (Hlawatsch et al., 2014; Vehlow et al., 2015). Sankey diagrams (Riehmann et al., 2005) and ThemeRiver (Havre et al., 2000) are the most common methods used. For example, Vehlow et al. (2015) used Sankey diagrams to show the changes of community structures. Flow-based techniques aggregate networks by group information, but lack details of the local areas of the network.

In this study, we combine adjacency matrices, Sankey diagrams, and tree structures based on a multi-focus tree cut algorithm and visualize the focused areas with fine-grained details and the unfocused areas with coarse-grained details within a sequence of matrices.

3 Overview

The RCD algorithms aim to find an initial example of rare classes in the data (Pelleg and Moore, 2005). To the best of our knowledge, batch-update incremental RCD (BIRD) (Zhou et al., 2015b) is the first (and only) work designed for detecting rare categories in dynamic networks. It takes snapshots of the dynamic network topology at two different time steps as input and iteratively detects rare category candidates, which potentially belong to a rare category. In this section, we first introduce the related concepts of BIRD, and then introduce the analytical tasks that users should complete based on RCAnalyzer to detect rare categories in dynamic networks.

3.1 Batch-update incremental rare category detection

Here, we review the key ideas of BIRD (Zhou et al., 2015b, 2017), which pave the way for our forthcoming introduction of the rare category visual analytic system. BIRD aims to detect rare categories in dynamic networks.

According to BIRD, a pair of nodes is closely connected if their transition probability is high. Therefore, BIRD believes that the transition probability of nodes in one rare category should have a lower bound and that the transition probability of nodes in different rare categories should have an upper bound (He et al., 2008). Therefore, a rare category is a group of connected nodes that possess the following two features: (1) These nodes form a compact structure, which means they are closely connected. The transition probabilities among these nodes are relatively high and larger than the lower bound. (2) The compact structure should have a clear border. The transition probabilities among the nodes in this structure (rare category) and the other rare categories are relatively low and smaller than the upper bound. There are two visual examples showing these two features intuitively in Fig. 2.

BIRD is an iterative algorithm. In each iteration, it detects a node whose neighborhood density significantly changes between two given adjacent time steps in a dynamic network. This node is potentially a representative node of a rare category.

Similar to the existing graph-based RCD algorithms (He and Carbonell, 2008; He et al., 2008; Zhou et al., 2015a), BIRD can be separated mainly into the following two parts:

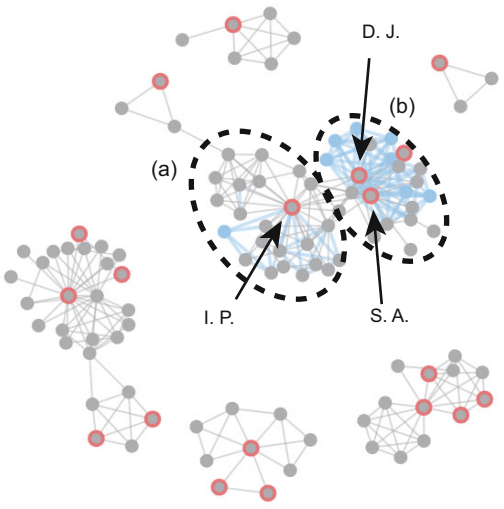

Fig. 2 Compact neighborhood structures of I. P. in area (a) and D. J. and S. A. in area (b)

1. Compute the global similarity matrix $\boldsymbol{A}$,

$$\boldsymbol{A} = (\boldsymbol{I} - \alpha \boldsymbol{W})^{-1}, \tag{1}$$

where $\boldsymbol{I}$ is an identity matrix, $\boldsymbol{W}$ denotes the transition probability matrix of the given graph G, and α is a positive discounting constant in the range of $(0, 1)$. Note that the global similarity matrix $\boldsymbol{A}$ helps sharpen the changes of the local density near the boundaries of each class. This considerably reduces the workload of identifying rare categories in the query process.

2. Update the query score iteratively based on the labeling information from users and return the example with the largest query score to users for inspection. In general, the query process selects the examples from regions where local density changes the most, and thus the queried examples tend to have a high possibility of hitting the regions of rare categories.

Before BIRD (Zhou et al., 2015b, 2017), previous studies (Pelleg and Moore, 2005; He and Carbonell, 2008; He et al., 2008, 2010) were all conducted for static graphs. For this reason, BIRD extends the problem to the dynamic setting and efficiently updates the RCD model using the local changes to avoid reconstructing it from scratch. To be specific, BIRD efficiently updates the global similarity matrix $\boldsymbol{A}^{(t)}$ at each time step t based on the global similarity matrix $\boldsymbol{A}^{(t-1)}$ at the previous time step $t-1$ and the updated edges at time step t, and locally updates the query scores of the examples which may be infected by the changes at time step t.

The original BIRD algorithm outputs the rare category candidates with the highest query score and waits for users to label the candidate. The query process might repeat many times. Thus, we slightly modify BIRD making the algorithm output candidates with the top k query scores, where k is a manually set parameter.

The workflow of analyzing rare categories in dynamic networks with BIRD contains three stages. First, users set parameters and select two adjacent snapshots to initialize BIRD. Second, users analyze and identify rare categories based on the candidates detected by BIRD. Third, users label the candidates, and the label result is returned to BIRD. When users think that all rare categories between the two snapshots are found, they can select other time steps and repeat the workflow to analyze other rare categories.

3.2 Analytical tasks

According to the analysis workflow, we summarize what analytical tasks should be completed by users based on the following:

1. Set parameters to initialize BIRD (T1). Users need to set a series of parameters before BIRD can detect rare category candidates. The most important parameters are the starting and ending time steps, which determine G_t used for initialization of BIRD.

2. Identify new rare categories from the examples detected by BIRD (T2). After BIRD is initialized, it will iteratively output the detected rare category candidates. Users first identify candidates that truly belong to rare categories by analyzing their neighborhood structure. Then users compare the detected rare category with the labeled rare categories to determine whether it is a new rare category.

3. Label the examples based on analysis results (T3). After analyzing rare category candidates, users label each candidate by a specific number. Labels are then returned to BIRD.

4 System design

In this section, we first introduce the design requirements of RCAnalyzer for completing the analytical tasks, and then introduce the design of RCAnalyzer in detail.

4.1 Design requirements

We identify the following design requirements that RCAnalyzer should fulfill based on the analytical tasks.

For setting parameters to initialize BIRD (T1), we identify the following design requirements:

1. Provide an overview of dynamic networks (R1). Users need to first explore the entire dynamic networks and understand the overall change of dynamic networks. With an overview, users can decide on which time periods they would focus.

To identify examples belonging to rare categories among all detected examples (T2), we identify the following design requirements:

2. Capture the changing structures of rare categories in the context of dynamic networks (R2). It is necessary to show the evolution of candidates in the background of the entire network. This helps users

identify the differences between the instance and the majority class.

3. Reveal the features of detected examples (R3). It is essential to show the features of the surrounding area of candidates to identify rare categories. The features include the ego network of the instance and the similar nodes detected by BIRD.

4. Reserve the context of labeled rare categories (R4). The system should remind users what kind of rare categories are detected and support the comparison between new candidates and labeled categories.

To label the examples based on analysis results (T3), we identify the following design requirements:

5. Enable users to set and reset the labels of candidates (R5). The system should enable users to label rare categories and change labels of rare categories when they make mistakes.

4.2 System pipeline

From the design requirements, we design the user interface of RCAnalyzer (Fig. 3). It consists of four parts: (1) the timeline view, which shows a high-level overview of dynamic networks (R1), (2) the matrices view, which shows the aggregated adjacency matrix of dynamic networks at each time segment initially (R1) and shows the details of the neighborhood of multiple vertices after node selection (R2), (3) the example view, which shows the feature of candidates (R3) and the query history of BIRD (R4), and (4) the label result view, which shows the historical label results and enables users to reset labels of labeled categories (R5 and R4).

Based on the analytical tasks, we design the architecture of RCAnalyzer (Fig. 4). RCAnalyzer consists of three major modules, a data storage module,

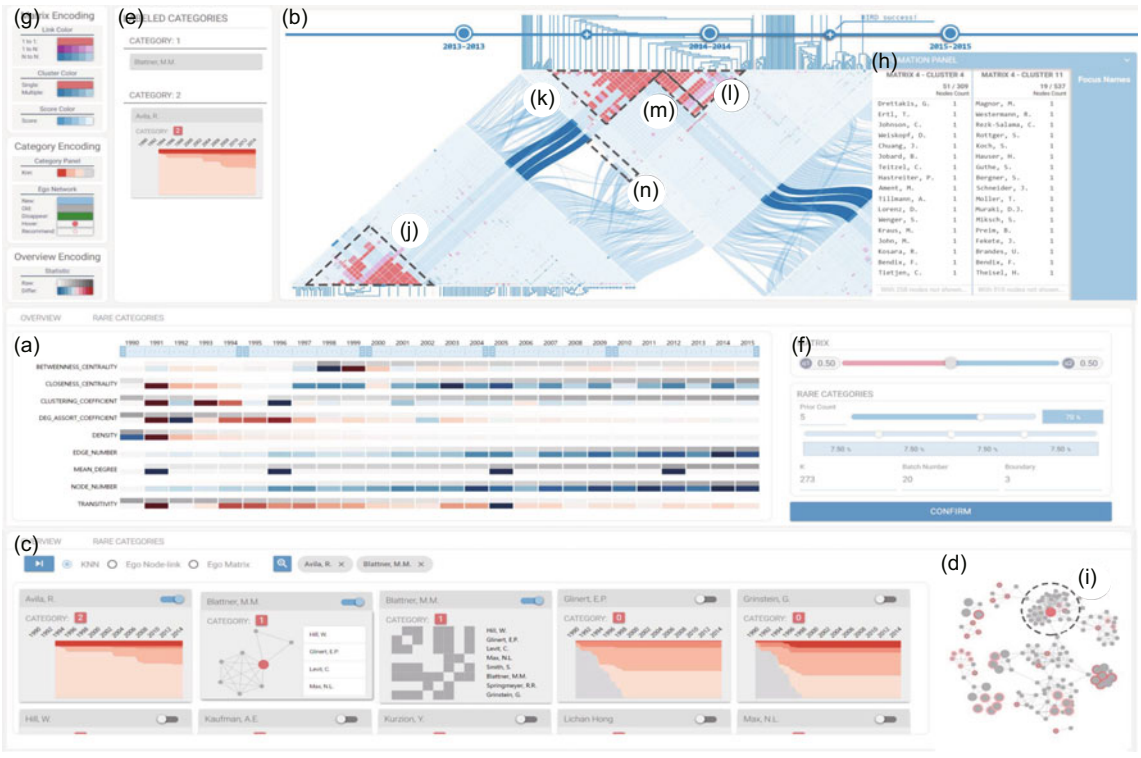

Fig. 3 User interface of RCAnalyzer: (a) time line view; (b) matrices view; (c) instance view; (d) sub-network view; (e) label result view; (f) parameter panel; (g) encoding panel; (h) information panel

BIRD detects W. D., X. W., and H. L. between 2014 and 2015. Area (i) is the compact neighborhood structures formed by them and their surrounding area in the sub-network view. Area (j) is the small community constituted by them and their surrounding areas in 2013. Area (k) is the same area as area (j) in 2014. Area (l) is a dense structure appearing beside area (k). Two nodes (area (m)) in area (k) have a lot of connections to nodes in area (l). Area (n) is the Sankey diagram which shows eight nodes in area (l) in 2014. Area (l) indicates the existence of a study with a lot of coauthors, which might be a result of multilateral cooperation. The abnormal changes of the surrounding areas of W. D., X. W., and H. L. make them a rare category. References to color refer to the online version of this figure

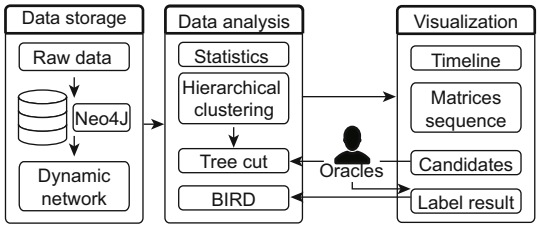

Fig. 4 System pipeline

a data analysis module, and a front-end visualization module. We use Neo4J to store the dynamic networks in the data storage module. The data analysis module contains four components, BIRD, a hierarchical clustering algorithm, a tree cut algorithm, and a bundle of statistics metrics. BIRD iteratively detects candidates of the rare categories. The hierarchical clustering algorithm extracts a tree structure from the network topology, and the tree cut algorithm groups nodes to clusters based on the tree structure and the network topology. The statistics metrics measure the macro condition of the dynamic network.

The visualization module contains four major views: (1) the timeline view, which shows the variation of network statistics and assists users in selecting, merging, and filtering time steps, (2) the matrices view, which visualizes the network dynamics based on the tree cut results, (3) the instance view, which displays the features of the rare category candidates detected by BIRD, and (4) the label result view, which reminds users what rare categories have been discovered.

4.3 Timeline view

The timeline view provides a highly abstracted overview of the dynamic network (R1). Metrics, including betweenness centrality, closeness centrality, clustering coefficient, degree assort coefficient, density, edge number, node number, average degree, and transitivity, are calculated to show the state of dynamic networks at each time stamp. The timeline view contains two parts, an interactive time axis and a pixel map. The pixel map visualizes metrics, which helps users find interesting snapshots of dynamic networks. The interactive time axis (Fig. 3a) enables users to select different snapshots (R1). After the time periods are submitted, the selected snapshots are accordingly extracted and merged. The data of merged snapshots are then visualized in the matrices view to show the network data in detail.

We consider using three different visual designs in the timeline view to visualize the metrics: a line chart, a pixel map, and a glyph design. A line chart is intuitive in showing time-varying data, while it lacks space efficiency. Using glyphs to individually show the metrics at each time stamp is space-efficient while lacking intuitiveness. Thus, we choose to use a pixel map to show the metrics because a pixel map is more space-efficient than a line chart and more intuitive than a series of glyphs.

4.4 Matrices view

After time periods are selected in the timeline view, the data analysis module first aggregates snapshots of the dynamic network according to the selected time periods. The matrices view is designed for showing the dynamics of the network topology and the dynamics of selected rare category candidates. A hierarchical clustering algorithm (Newman and Girvan, 2004), which builds a dendrogram based on network topology, is applied on each aggregated snapshot to reduce the number of entries in each matrix, because a large matrix can hardly be visualized in a limited space with satisfactory details. Same clusters at different time stamps are linked to show the dynamics of the network. However, users cannot really explore or compare the neighborhood of rare category candidates in aggregated matrices because of the lack of details. Therefore, a multi-focus tree cut algorithm is applied to each dendrogram to provide fine-grained details of user-selected candidates and coarse-grained details of other nodes. In this way, users can observe and compare the evolution pattern of rare category candidates (R2).

4.4.1 Multi-focus tree cutting

When users are interested in one or more rare category candidates, the dynamics of neighborhoods of these candidates are shown in the matrices view to support users in exploring, comparing, and identifying rare categories among these candidates. We design a multi-focus tree cut algorithm to enable the matrices view to provide fine-grained details around the selected nodes and coarse-grained details around the unrelated nodes, which supports users in identifying rare categories among candidates (T2) by comparing the features of candidates, labeled rare

categories, and non-rare categories. Different from existing "multi-focus+context" approaches (Gansner et al., 2005; Feng et al., 2012; Sundararajan et al., 2013), which work on the layout results of networks, our method directly works on the network topology and thus does not depend on the layout of networks.

Suppose we are given a dynamic network, which consists of a series of snapshots, $\mathbb{G} = \{G^1, G^2, \ldots, G^t\}$. The multi-focus tree cutting algorithm works on each snapshot. The algorithm consists of two stages. In the first stage, details around all focused nodes are cut out from the tree. In the second stage, a merge operation is applied to prevent the results from containing too many non-relevant single-node clusters.

1. First stage: multi-focus tree cutting

The procedure of the first stage is shown in Fig. 5. For a specific snapshot $G^i = (V, E)$, hierarchical clustering is applied first to obtain a tree structure based on modularity (Newman and Girvan, 2004). To cut the tree with multiple focused nodes, we modify the original modularity. The set of focused nodes can be written as $F = \{n|\text{focused nodes}\}$. The cut of the tree structure is an optimization of an energy function based on the tree structure and the network topology. Suppose the cutting result is $C = \{N_1, N_2, \ldots, N_m\}$, where N_i is a group of nodes in the tree.

$$C = \arg\min \sum_{i=1}^{m} E(N_i), \tag{2}$$

where

$$\begin{cases} E(N_i) = \sum_{e \in N_i} \left(\dfrac{D(e, N_i)}{|N_i|} - \left(\dfrac{S(e, N_i)}{|N_i|} \right)^2 \right), \\ D(e, N_i) = \begin{cases} \text{Weight}(e), & \text{if } \forall v \in e, v \in N_i, \\ 0, & \text{otherwise}, \end{cases} \\ S(e, N_i) = \begin{cases} \text{Weight}(e), & \text{if } \exists v \in e, v \in N_i, \\ 0, & \text{otherwise}. \end{cases} \end{cases} \tag{3}$$

We define the weight of an edge as the minimum value of the weights of the node: supposing $e = (v_1, v_2)$, then $\text{Weight}(e) = \min(\text{Weight}(v_1), \text{Weight}(v_2))$. The weight of a node is defined based on the distance between the node and the focus node in the tree structure or the network topology:

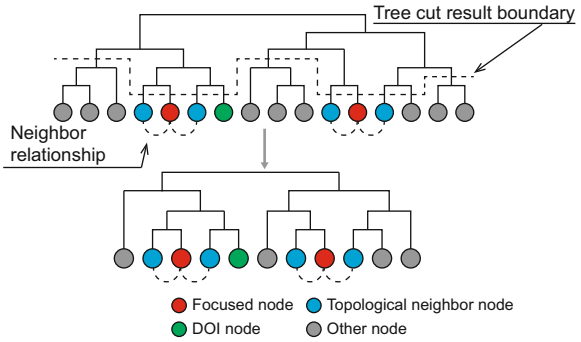

Fig. 5 First stage of the tree cut algorithm: keeping the details of all focused nodes

$$\begin{cases} \text{Weight}(v) = \alpha_1 W_{\text{DOI}}(v) + \alpha_2 W_{\text{Topology}}(v), \\ W_{\text{DOI}}(v) = \min\limits_{n \in F}(D_{\text{DOI}}(n, v)), \\ W_{\text{Topology}}(v) = \min\limits_{n \in F}(D_{\text{Topology}}(n, v)), \end{cases} \tag{4}$$

where $D_{\text{DOI}}(n, v)$ is the degree of interest distance between n and focused node v in the tree structure, $D_{\text{Topology}}(n, v)$ is the shortest distance between n and v in the network topology, and α_1 and α_2 are weights of $D_{\text{DOI}}(n, v)$ and $D_{\text{Topology}}(n, v)$, respectively.

2. Second stage: re-clustering of non-relevant nodes in the partial structure

When the structure of a hierarchical clustering tree is partial and the focused nodes are deep in the tree, a large number of non-relevant nodes might be cut out from the tree, which increases the height of the cut result.

To avoid this problem, we apply a re-cluster procedure to the non-relevant nodes. The continuous non-relevant single node sequences are first detected and cut out from the tree. Then the tree cut algorithm is applied again to the sub-tree based on the network topology. Last, hierarchies are inserted back into the tree. The procedure of this stage is shown in Fig. 6.

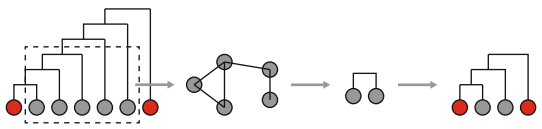

Fig. 6 Second stage of the tree cut algorithm: regrouping the unrelated nodes according to the network structure

4.4.2 Visual design in the matrices view

We use a combination of matrix, Sankey diagrams, and dendrogram as the basic representation of dynamic networks (Fig. 1). Sankey diagrams are added between each pair of adjacent matrices to show the evolution of these groups. The hierarchy of clusters represents the relationships among clusters and the structure of the network. In RCAnalyzer, all networks are treated as undirected networks, and thus the adjacent matrices are symmetric. We use dendrograms to replace the upper (lower) triangular matrices and show the hierarchy of clustering results for space efficiency. The sequence of upper and lower triangular matrices is laid in a zigzag shape (Fig. 1).

Due to the tree cut algorithm, there are different granularity details, leading to different numbers of nodes in different clusters. The opacity and color of triangles on the diagonal of matrices encode the number of nodes (Fig. 7). We use blue and red to distinguish a group of nodes and a single node. The gradient of blue in Fig. 7 is used to encode the number of nodes in groups. Rectangles inside matrices represent three categories of connections: a single node to a single node, a single node to a group of nodes, and a group of nodes to a group of nodes. For consistency, we use blue to encode

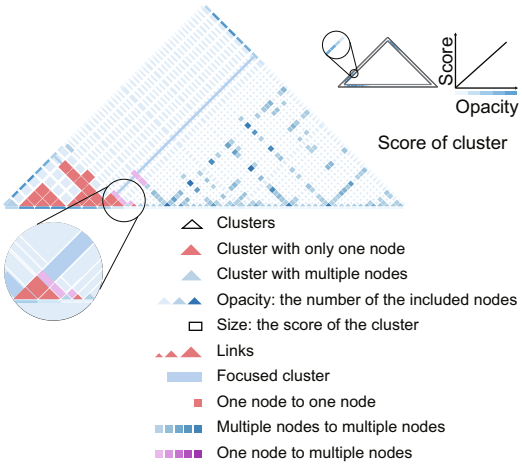

Fig. 7 Visual encodings inside a matrix

Triangles represent a single node (red) or a group of nodes (gradient blue showing size). A red rectangle represents the connection between two single nodes. A purple rectangle represents the connections between a single node and a group of nodes. A blue rectangle represents the connections between two groups of nodes. Scores are encoded by both the sizes of rectangles and triangles and the color on the matrix border. References to color refer to the online version of this figure

group-to-group relations, orange to encode one-to-one relations, and purple to encode one-to-group relations. The gradient of colors represents the actual number of connections between the corresponding nodes.

Due to the importance of node anomalies in this study, we decide to use the size of triangles on the diagonal of matrices to encode the anomalous scores output by BIRD (R3). If a large number of clusters are generated by the tree cut algorithm, sizes of single node clusters will be small under the limited size of matrices, which impedes the analysis of the nodes in which users are interested. We use three methods simultaneously to solve this problem. First, freely zooming and dragging are supported in this view. When the matrices are enlarged, the sequence of matrices cannot be fully displayed because of the limitation of space. Thus, we implement a special scale interaction with the scale functions (Fig. 8) to enable local scaling without changing the size of matrices.

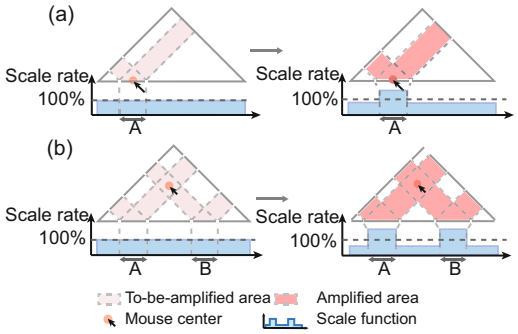

Fig. 8 Scale functions: (a) focus on border—single amplified area; (b) general condition—two amplified areas

When scale interaction is activated, the distortion of the size of the triangles and rectangles may mislead users, although we maintain the size ratio in the scaled local area. Thus, we encode the scores on the borders of the matrices by colors, which brings two benefits: (1) Users will clearly distinguish to which clusters the bands in Sankey diagrams belong when matrices are sparse. (2) Users will observe the changes of scores over time stamps more easily.

Node-link diagram and matrix representation are two common techniques to visualize networks. We choose the matrix as the basic representation of networks instead of the node-link diagram because the matrix representation can be better combined

with a dendrogram. Although the same clusters or nodes can be linked in a series of node-link diagrams to visualize a dynamic network, overlap of lines in this solution will be severe and significantly reduce the readability of the visualization.

4.5 Rare category candidates view

The rare category candidates view is designed to reveal the features of candidates (R3). It contains two components: small-multiples of candidate feature panels, which visualize the neighborhood information of candidates, and a sub-network view, which shows the sub-network formed by all detected candidates and their first-hop friends.

Representation of the ego network of candidates consists of two visualization forms: a node-link diagram and a matrix. The coexistence of node-link diagrams and matrices is not considered redundant because we think the two visualization forms have different emphases. The former emphasizes vertices, while the latter emphasizes links. Because BIRD detects rare categories between two time steps, changes of the candidates' ego networks at the two time steps are as shown in Fig. 3. The states of vertices and links are encoded by colors: blue indicates appearance, green indicates disappearance, and grey indicates fixedness.

The sub-network of candidates shows the query process of BIRD by visualizing all the candidates together with their first-hop-neighbors (R3) and helps users compare the candidates in the local area of the network. Color encoding is similar to encoding in ego networks. Except for the color of links and nodes, we use the red border of nodes to demonstrate the candidates detected in the current iteration and the light red border of nodes to demonstrate the candidates detected in previous iterations. When an instance is hovered, both itself and its k nearest neighbors will be enlarged (Fig. 9).

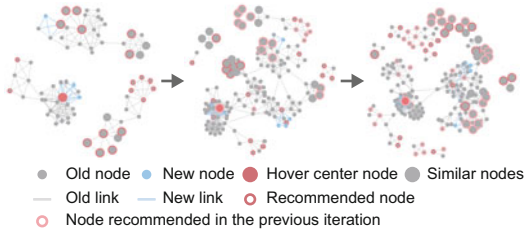

Fig. 9 Query process of BIRD by a node-link diagram formed by all the candidates ever queried by BIRD

4.6 Other panels

The label result view shows the rare categories detected by recording label results of rare category candidates in a list of candidate feature panels (R4), as shown in Fig. 3. Users can review the detected rare categories at any time during the analysis procedure.

The encoding panel shows the color encodings used in the system (Fig. 3g). The information panel shows the detailed information of the selected blocks in the matrices view, as shown in Fig. 3h. When hovering on triangles on the diagonal of a matrix, the node count and node list are shown in the panel. When hovering on rectangles inside a matrix, the information panel is divided into two parts, each of which shows the node count and the nodes that have connections to the other cluster. The link count between two clusters is also shown in Fig. 3h.

4.7 User interaction

The system implements a series of user interactions to support users in analyzing the rare categories.

1. Details on demand

The instance view and the matrices view show the information of rare candidates at different levels of details. Once nodes are selected in the instance view, the tree cut algorithm will be applied and the detailed information of the selected candidates and their related nodes will be shown in the matrices sequence view with the context of the entire dynamic network.

2. Highlighting and pinning

All views in RCAnalyzer are linked. Whenever and wherever a node is hovered over by users, other views will highlight the node and its related nodes. Users can pin the block by clicking on it and then explore the details in the information panel.

3. Dragging and zooming

The matrices view supports users in freely dragging and zooming the matrix sequence.

4. Rare category labeling

Users can label each candidate with a specific number, which helps BIRD distinguish different rare categories in the feature panel.

5 System evaluation

In this section, we conduct one use scenario and a controlled user study to demonstrate the effectiveness of RCAnalyzer. The use scenario is based on a dynamic network extracted from the collaboration among authors of visualization publications (Isenberg et al., 2017).

We have developed a prototype system to do all the experiments. RCAnalyzer is a web application which supports multiple users in analyzing the rare categories in dynamic networks. The front-end visualization is implemented by AngularJS, D3, and CSS. The back-end server is implemented by Python with Flask, Neo4J, numpy, igraph, and networkx. The use scenario and user study run on a computer with Intel® Core™ i7-4770 CPU and 20 GB RAM.

5.1 Use scenario: collaboration network in visualization publications

We have extracted all co-authorships in the IEEE VIS dataset (Isenberg et al., 2017) from 1990 to 2015. An incremental collaboration network was constructed based on co-authorships, in which a link at time stamp t indicates that two authors have coauthored at t or before t. We filtered the authors by taking the largest connected component in 2015, and there are 3640 authors left in the network. The number of links varies from 43 (1990) to 11 848 (2015).

The timeline view and matrices view show the basic information of the network (Figs. 3a and 3b). Note that the time axis is initially divided into five segments to show the condition of the dynamic network in the periods of time. The heatmap and the matrices show that before 2000, both the number and the increment of nodes and links are small. After 2000, the network grows faster, and after 2004, the network significantly grows.

After initializing BIRD with the data in 2014 and 2015, W. D., X. W., and H. L. are selected to be the focused nodes in the instance view, as shown in Fig. 3. They and their neighbors form a compact area in the sub-network view (area (i) in Fig. 3). Their surrounding areas from 2013 to 2015 are shown in the matrices view. Focused nodes are highlighted by the blue lines. Area (j) in Fig. 3 is their surrounding area in 2013. The large link density in this area indicates that nodes in this area have close collaboration relationships. Thus, these nodes can be regarded as a small collaboration group. The Sankey diagram between 2013 and 2014 shows that area (k) is almost the same as area (j). A dense structure in area (l) appears beside area (k). Meanwhile, area (m) shows that two nodes, including X. W. in area (k), connect to most nodes in area (l). The blank of the Sankey diagram (labeled by area (n)) on the left of the matrix in 2014 indicates that eight nodes in area (l) are new nodes. The clique structure in area (l) indicates that these nodes collaborate in the same study. A large number of authors of the study indicate that the study might be the result of multilateral cooperation. The appearance of this uncommon cooperation causes W. D., X. W., and H. L. to be identified as a rare category.

Between 2012 and 2013, D. J., S. A., and I. P. constitute a large and dense sub-network (Fig. 2). However, there is a small gap between the first two authors (area (b) in Fig. 2) and the last author (area (a) in Fig. 2). Thus, whether they belong to the same category cannot be decided. The matrices view shows the dynamic changes in surrounding areas around them (Fig. 10). In 2011, I. P. was in area (a) in Fig. 10, and D. J. and S. A. were in area (b) in Fig. 10. It is clear that these two areas have no connections. In 2012, area (c) in Fig. 10 shows that the two areas in 2011 merged into one because of the new connections in area (d). However, a large number of new connections appeared in area (e) in 2013. From the Sankey diagram between 2013 and 2014, we know that authors newly connected to D. J. and S. A. in 2013 also appeared in area (g) in 2014. From the matrix of 2014, we can see that areas (g) and (h) were separated from each other. Thus, the merging and splitting behaviors of the surrounding areas of D. J., S. A., and I. P. along time are the reasons why D. J., S. A., and I. P. are identified as a rare category.

5.2 User study

We have conducted a user study to verify the usability of RCAnalyzer. We introduced the user study following the order of assumptions, synthetic datasets, participants, tasks, procedures, results, and qualitative feedbacks.

1. Assumptions

As there is no existing work supporting similar tasks to RCAnalyzer, we did not use a baseline system in this user study and tested only whether RCAnalyzer could help users explore, analyze, and

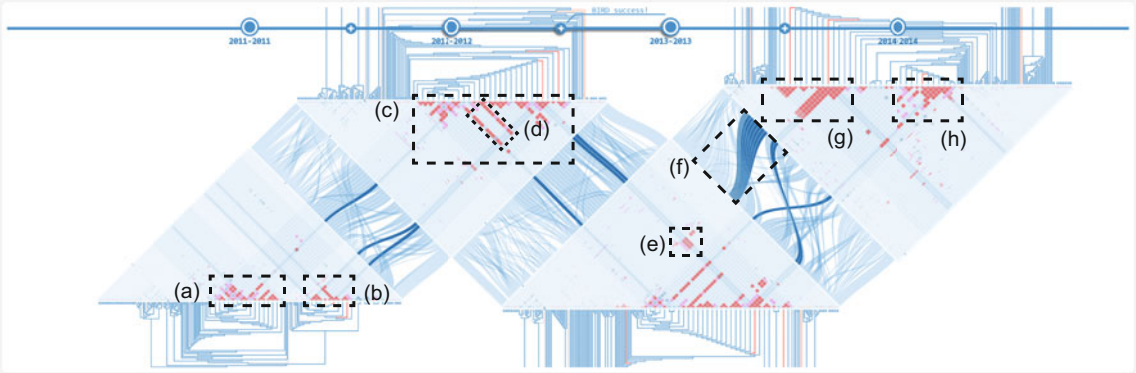

Fig. 10 Dynamic changes in surrounding areas around D. J., S. A., and I. P.

In 2011, the surrounding area of I. P. (area (a)) and the surrounding area of D. J. and S. A. (area (b)) were not connected. The links appearing in 2012 (area (d)) connected areas (a) and (b) to area (c). In 2013, a large number of links appeared. The Sankey diagram shows nodes connected by the links form a dense structure with D. J. and S. A. in 2014 (area (g)). The newly appearing links make areas (g) and (h) two separated categories

identify rare categories in dynamic networks and collect users' qualitative feedbacks. We first make three assumptions about the usability of RCAnalyzer: (1) RCAnalyzer helps users identify examples of rare categories among the query result of BIRD in each iteration. (2) RCAnalyzer helps users distinguish examples of rare categories and examples of major categories. (3) RCAnalyzer helps users distinguish examples of different rare categories.

For a dataset with ground truth, we can count the minimum number of iterations within which BIRD can detect at least one example in each of the rare categories in the dataset. By comparing this minimum number and the actual number of iterations, we can validate assumption (1). If the number of iterations used by users is close to the minimum number, RCAnalyzer efficiently supports users in identifying rare categories. We validated assumptions (2) and (3) by calculating the accuracy of the rare categories labeled by users in this user study.

2. Synthetic datasets

Because of the high complexity of the real datasets used in the case study, it is hard to control the test and quantify the actual efficiency of RCD with RCAnalyzer. Thus, we used synthetic datasets in the user study. All the synthetic datasets had two time stamps. Each synthetic dataset was constructed by the following procedure: (1) generating a grid network with N nodes at each time stamp; (2) adding edges among nodes in the network to form four different special structures (a clique, a bipartite graph, a star, and a circle) at the second time stamp. Special structures are treated as rare categories and other nodes as the major category. We constructed four synthetic datasets with $N = 100$, 200, 500, and 1000. The dataset with $N = 100$ was used in the tutorial of the user study. The minimum numbers of iterations on datasets with $N = 200$, 500, and 1000 are 5, 5, and 11, respectively.

3. Participants

We recruited 12 participants for the evaluation, including nine males and three females. All of them have background in visualization, and one of them has a background in anomaly detection.

4. Tasks

The participants were asked to complete the following tasks in this user study: (1) identifying rare categories in the examples detected by BIRD in each iteration; (2) labeling examples identified as rare categories.

5. Procedures

The user study had three stages. In the first stage, we introduced the basic concept of this work and the tasks of the user study to participants with a 10-min tutorial. In the second stage, we introduced RCAnalyzer to participants and let them explore the system with the synthetic dataset with $N = 100$ for 15 min. Participants were allowed to ask any question about the system and the tasks in the first and second stages. In the third stage, participants were asked to analyze the synthetic datasets with $N = 200$, 500, and 1000, label rare categories they identified in RCAnalyzer, and write down their labeling results on an answer sheet. To ensure that

participants will not arbitrarily give answers, they were asked to describe the reason why a detected example is identified as a rare category.

6. Results

The accuracy of labeling rare categories is shown in Table 1. The results show that the detection of the clique, bipartite graph, and star is accurate (86.11%, 86.11%, and 91.67%), while the accuracy of detection of the circle is not very good (77.87%). Detection of the circle is really hard because the surrounding area of a node on the circle is unobtrusive in the matrices view and nodes on the circle are queried by BIRD discontinuously, forming several segments of line instead of a circle in the sub-network view. To identify the circle structure, participants need to select a series of instances on the circle, but some of the participants missed too many instances on the circle, and thus were not able to label the circle structure correctly. The distribution of participants' query number is shown in Fig. 11, which shows that participants can finish the labeling process in 4–5 iterations in datasets with 200 and 500 nodes. For the dataset with 1000 nodes, many of the participants can finish the labeling process in 11–15 iterations, while two outliers finished the labeling process in two and six iterations, respectively. This is because they labeled normal nodes as rare categories. Some of the participants finished the labeling after over 20 iterations. This is because they did not label the rare categories promptly. Overall, the accuracy and the average query numbers show that most of the participants can identify rare categories promptly and correctly.

Table 1 Accuracy of labeling for four different special structures

Number of nodes	Accuracy			
	Clique	Bipartite graph	Star	Circle
200	91.67%	83.33%	83.33%	58.33%
500	83.33%	83.33%	100.00%	91.67%
1000	83.33%	91.67%	91.67%	83.33%
Average	86.11%	86.11%	91.67%	77.78%

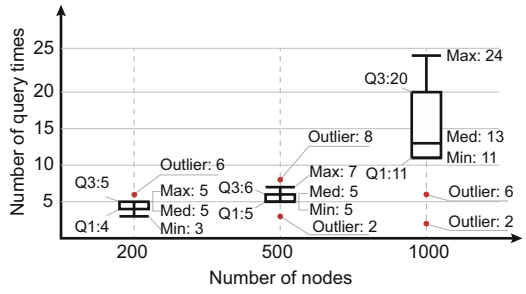

Fig. 11 Query numbers of participants when labeling all rare categories in the datasets

7. Qualitative feedbacks

To assess the learnability, usability, and other perception aspects of RCAnalyzer, users were asked to give some qualitative feedbacks after the formal user study. The most frequent complaint was that the encodings in our matrices view were too complex. We used both the size and color of each cell to encode different information. Users had to recognize all the encodings at the beginning of the user study. It would lead to confusion because they would forget the encodings. Some users said that the parameters were hard to comprehend. They said that it was hard to learn what would happen if the parameters were adjusted. It took a long time for them to learn how the system works. Learnability and usability were both important problems which are hard to cover. One of the solutions for improvement is to reduce the complexity of our visual design. However, it takes much more time to know which visual design is less efficient and can be abandoned. In the future, we will improve our visual design based on more user behaviors. For example, color encoding on the border can be removed if users do not care about border color encoding.

6 Discussions

1. Generalizability

Although we support only BIRD in our system, RCAnalyzer can work based on other RCD algorithms as long as they are based on the topology of dynamic networks. The matrices view with the tree cut algorithm can be applied in other applications for analyzing dynamic networks, for example, tracking the time-varying pattern of multiple nodes and comparing the change of ego networks of multiple nodes. We believe that the combination of the matrix sequence and the multi-focus tree cut algorithm is a useful method as it enables simultaneous comparison of multiple nodes.

2. Scalability

In the use scenario, we tested the effectiveness of RCAnalyzer on a network with 8319 nodes, 210–625 edges, and 6 time steps, indicating that RCAnalyzer has good scalability on large datasets. As for

larger datasets, the major bottleneck would be the running time of the initialization of BIRD and the tree cut algorithm due to the limitation of execution efficiency of Python. In the future, we plan to use pre-computation and server-side cache to support the analysis of larger datasets. The scalability of our visual design is related to the granularity of our tree cut algorithm and the scale of the input dynamic network. From our experience, it is hard to show more than six time steps with around 50 rows in each matrix at the same time in the matrices view (with 1360 × 635 pixels). Interactions such as dragging and zooming to improve the readability of matrices have been discussed in Section 4.4.2. For dynamic graphs with more time steps, the tree cut algorithm should be more coarse-grained to show all time steps in the meantime. However, the coarse-grained tree cut algorithm reduces the information of the dynamic networks.

3. Limitations

Although RCAnalyzer can help users analyze and label rare categories in dynamic networks, it still has several limitations. First, more interactions should be supported, such as querying and filtering. Interactions in RCAnalyzer are enough to support the detection of rare categories, but more complete interactions can significantly improve user experience. Second, the processes of interactions and visual encodings in RCAnalyzer are a little complicated. During the user study, it takes 15–25 min to train subjects to let them fully understand how to use the system. Third, RCAnalyzer supports only screens with 1920 × 1080 resolution. More adaptive layout should be supported to enable users to label rare categories at different resolutions.

4. Future work

First, we plan to add the context information of nodes in RCAnalyzer. RCAnalyzer is currently based on the topology of dynamic networks because BIRD detects rare categories by checking the changes of the topological structure around nodes. However, nodes with the same topology may have completely different context information. We believe that the context information will help users distinguish different rare categories. Second, we plan to add data filtering to RCAnalyzer. Sometimes, users might be interested in only a special area in the network. A data filtering module can help them analyze the desired areas of data.

7 Conclusions

In this paper, we present RCAnalyzer, a novel visual analytics system which helps oracles analyze the results of RCD methods and label rare categories in dynamic networks. It consists of five linked views (timeline view, matrices view, instance view, subnetwork view, and label result view), and shows the information of rare categories at different levels of details. In addition, we present a multi-focus tree cut algorithm and a tree-structure constrained layout optimization algorithm to support the comparison of instances in the context of their surrounding structures. We use a use scenario and a user study to demonstrate the usability and effectiveness in analyzing rare categories in dynamic networks.

Contributors

Dong-ming HAN processed the data and designed the experiment. Fang-zhou GUO designed the research and drafted the manuscript. Da-wei ZHOU provided the RCD algorithm. Nan CAO and Wei CHEN helped organize the manuscript. Jing-rui HE and Ming-liang XU polished the paper. Jia-cheng PAN implemented the interface and finalized the paper.

Compliance with ethics guidelines

Jia-cheng PAN, Dong-ming HAN, Fang-zhou GUO, Da-wei ZHOU, Nan CAO, Jing-rui HE, Ming-liang XU, and Wei CHEN declare that they have no conflict of interest.

The Ethics Committee of Zhejiang University had reviewed the experimental procedure and method, and approved this experiment. Before the experiment, all subjects signed the informed written consent and agreed to participate in this experiment.

References

Archambault D, Purchase H, Pinaud B, 2011. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Trans Vis Comput Graph*, 17(4):539-552. https://doi.org/10.1109/TVCG.2010.78

Bach B, Pietriga E, Fekete JD, 2014a. GraphDiaries: animated transitions and temporal navigation for dynamic networks. *IEEE Trans Vis Comput Graph*, 20(5):740-754. https://doi.org/10.1109/TVCG.2013.254

Bach B, Pietriga E, Fekete JD, 2014b. Visualizing dynamic networks with matrix cubes. SIGCHI Conf on Human Factors in Computing Systems, p.877-886. https://doi.org/10.1145/2556288.2557010

Bach B, Henry-Riche N, Dwyer T, et al., 2015. Small MultiPiles: piling time to explore temporal patterns in dynamic networks. *Comput Graph Forum*, 34(3):31-40. https://doi.org/10.1111/cgf.12615

Beck F, Burch M, Diehl S, et al., 2014. The state of the art in visualizing dynamic graphs. Eurographics Conf on Visualization, p.1-21. https://doi.org/10.2312/eurovisstar.20141174

Bhuyan MH, Bhattacharyya DK, Kalita JK, 2014. Network anomaly detection: methods, systems, and tools. *IEEE Commun Surv Tutor*, 16(1):303-336. https://doi.org/10.1109/SURV.2013.052213.00046

Blanch R, Dautriche R, Bisson G, 2015. Dendrogramix: a hybrid tree-matrix visualization technique to support interactive exploration of dendrograms. Proc IEEE Pacific Visualization Symp, p.31-38. https://doi.org/10.1109/PACIFICVIS.2015.7156353

Brandes U, Nick B, 2011. Asymmetric relations in longitudinal social networks. *IEEE Trans Vis Comput Graph*, 17(12):2283-2290. https://doi.org/10.1109/TVCG.2011.169

Burch M, Schmidt B, Weiskopf D, 2013. A matrix-based visualization for exploring dynamic compound digraphs. 17th Int Conf on Information Visualisation, p.66-73. https://doi.org/10.1109/IV.2013.8

Cao N, Gotz D, Sun JM, et al., 2011. DICON: interactive visual analysis of multidimensional clusters. *IEEE Trans Vis Comput Graph*, 17(12):2581-2590. https://doi.org/10.1109/TVCG.2011.188

Cao N, Shi C, Lin S, et al., 2016. TargetVue: visual analysis of anomalous user behaviors in online communication systems. *IEEE Trans Vis Comput Graph*, 22(1):280-289. https://doi.org/10.1109/TVCG.2015.2467196

Chandola V, Banerjee A, Kumar V, 2009. Anomaly detection: a survey. *ACM Comput Surv*, 41(3):15. https://doi.org/10.1145/1541880.1541882

Corchado E, Herrero Á, 2011. Neural visualization of network traffic data for intrusion detection. *Appl Soft Comput*, 11(2):2042-2056. https://doi.org/10.1016/j.asoc.2010.07.002

Fan X, Li CL, Yuan XR, et al., 2019. An interactive visual analytics approach for network anomaly detection through smart labeling. *J Vis*, 22(5):955-971. https://doi.org/10.1007/s12650-019-00580-7

Feng KC, Wang CL, Shen HW, et al., 2012. Coherent time-varying graph drawing with multifocus+context interaction. *IEEE Trans Vis Comput Graph*, 18(8):1330-1342. https://doi.org/10.1109/TVCG.2011.128

Gansner ER, Koren Y, North SC, 2005. Topological fisheye views for visualizing large graphs. *IEEE Trans Vis Comput Graph*, 11(4):457-468. https://doi.org/10.1109/TVCG.2005.66

Haberkorn T, Koglbauer I, Braunstingl R, 2014. Traffic displays for visual flight indicating track and priority cues. *IEEE Trans Human Mach Syst*, 44(6):755-766. https://doi.org/10.1109/THMS.2014.2352496

Havre S, Hetzler B, Nowell L, 2000. ThemeRiver: visualizing theme changes over time. IEEE Symp on Information Visualization, p.115-123. https://doi.org/10.1109/INFVIS.2000.885098

He JR, Carbonell JG, 2008. Nearest-neighbor-based active learning for rare category detection. 20th Int Conf on Neural Information Processing Systems, p.633-640.

He JR, Carbonell JG, 2009. Prior-free rare category detection. SIAM Int Conf on Data Mining, p.155-163.

He JR, Liu Y, Lawrence R, 2008. Graph-based rare category detection. 8th IEEE Int Conf on Data Mining, p.833-838. https://doi.org/10.1109/ICDM.2008.122

He JR, Tong HH, Carbonell JG, 2010. Rare category characterization. Proc IEEE Int Conf on Data Mining, p.226-235. https://doi.org/10.1109/ICDM.2010.154

Heard NA, Weston DJ, Platanioti K, et al., 2010. Bayesian anomaly detection methods for social networks. *Ann Appl Stat*, 4(2):645-662. https://doi.org/10.1214/10-AOAS329

Henry N, Fekete JD, McGuffin MJ, 2007. NodeTrix: a hybrid visualization of social networks. *IEEE Trans Vis Comput Graph*, 13(6):1302-1309. https://doi.org/10.1109/TVCG.2007.70582

Hlawatsch M, Burch M, Weiskopf D, 2014. Visual adjacency lists for dynamic graphs. *IEEE Trans Vis Comput Graph*, 20(11):1590-1603. https://doi.org/10.1109/TVCG.2014.2322594

Huang H, He QM, He JF, et al., 2011. RADAR: rare category detection via computation of boundary degree. Proc 15th Pacific-Asia Conf on Advances in Knowledge Discovery and Data Mining, p.258-269. https://doi.org/10.1007/978-3-642-20847-8_22

Huang H, He QM, Chiew K, et al., 2013. CLOVER: a faster prior-free approach to rare-category detection. *Knowl Inform Syst*, 35(3):713-736. https://doi.org/10.1007/s10115-012-0530-9

Inselberg A, 2009. Parallel Coordinates: Visual Multidimensional Geometry and its Applications. Springer, New York, USA. https://doi.org/10.1007/978-0-387-68628-8

Isenberg P, Heimerl F, Koch S, et al., 2017. Vispubdata.org: a metadata collection about IEEE visualization (VIS) publications. *IEEE Trans Vis Comput Graph*, 23(9):2199-2206. https://doi.org/10.1109/TVCG.2016.2615308

Jolliffe, IT, 1986. Principal Component Analysis. Springer, Berlin, Germany.

Jovanovic J, Bagheri E, Gasevic D, 2015. Comprehension and learning of social goals through visualization. *IEEE Trans Human Mach Syst*, 45(4):478-489. https://doi.org/10.1109/THMS.2015.2419083

Lin HF, Gao SY, Gotz D, et al., 2018. RCLens: interactive rare category exploration and identification. *IEEE Trans Vis Comput Graph*, 24(7):2223-2237. https://doi.org/10.1109/TVCG.2017.2711030

Liu Y, Dai S, Wang C, et al., 2017. GenealogyVis: a system for visual analysis of multidimensional genealogical data. *IEEE Trans Human Mach Syst*, 47(6):873-885. https://doi.org/10.1109/THMS.2017.2693236

Newman MEJ, Girvan M, 2004. Finding and evaluating community structure in networks. *Phys Rev E*, 69(2): 026113. https://doi.org/10.1103/PhysRevE.69.026113

Oelke D, Kokkinakis D, Keim DA, 2013. Fingerprint matrices: uncovering the dynamics of social networks in prose literature. *Comput Graph Forum*, 32(3pt4):371-380. https://doi.org/10.1111/cgf.12124

Pelleg D, Moore AW, 2005. Active learning for anomaly and rare-category detection. Proc 17th Int Conf on Neural Information Processing Systems, p.1073-1080.

Ranshous S, Shen ST, Koutra D, et al., 2015. Anomaly detection in dynamic networks: a survey. *WIREs Comput Stat*, 7(3):223-247. https://doi.org/10.1002/wics.1347

Riehmann P, Hanfler M, Froehlich B, 2005. Interactive Sankey diagrams. IEEE Symp on Information Visualization, p.233-240. https://doi.org/10.1109/INFVIS.2005.1532152

Sun JM, Faloutsos C, Papadimitriou S, et al., 2007. GraphScope: parameter-free mining of large time-evolving graphs. Proc 13th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.687-696. https://doi.org/10.1145/1281192.1281266

Sundararajan PK, Mengshoel OJ, Selker T, 2013. Multi-focus and multi-window techniques for interactive network exploration. SPIE Electronic Imaging, p.282-296. https://doi.org/10.1117/12.2005659

Teoh ST, Ma KL, Wu SF, et al., 2002. Case study: interactive visualization for Internet security. Proc Conf on IEEE Visualization, p.505-508. https://doi.org/10.1109/VISUAL.2002.1183816

Thom D, Bosch H, Koch S, et al., 2012. Spatiotemporal anomaly detection through visual analysis of geolocated Twitter messages. Proc Pacific Visualization Symp, p.41-48. https://doi.org/10.1109/PacificVis.2012.6183572

Tsai CF, Hsu YF, Lin CY, et al., 2009. Intrusion detection by machine learning: a review. *Expert Syst Appl*, 36(10):11994-12000. https://doi.org/10.1016/j.eswa.2009.05.029

van den Elzen S, Holten D, Blaas J, et al., 2016. Reducing snapshots to points: a visual analytics approach to dynamic network exploration. *IEEE Trans Vis Comput Graph*, 22(1):1-10. https://doi.org/10.1109/TVCG.2015.2468078

Vatturi P, Wong WK, 2008. Category detection using hierarchical mean shift. Proc 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.847-856. https://doi.org/10.1145/1557019.1557112

Vehlow C, Beck F, Auwärter P, et al., 2015. Visualizing the evolution of communities in dynamic graphs. *Comput Graph Forum*, 34(1):277-288. https://doi.org/10.1111/cgf.12512

Wang C, Xiao Z, Liu Y, et al., 2013. SentiView: sentiment analysis and visualization for Internet popular topics. *IEEE Trans Human Mach Syst*, 43(6):620-630. https://doi.org/10.1109/THMS.2013.2285047

Xu PP, Mei HH, Liu R, et al., 2017. ViDX: visual diagnostics of assembly line performance in smart factories. *IEEE Trans Vis Comput Graph*, 23(1):291-300. https://doi.org/10.1109/TVCG.2016.2598664

Yee KP, Fisher D, Dhamija R, et al., 2001. Animated exploration of dynamic graphs with radial layout. IEEE Symp on Information Visualization, p.43-50. https://doi.org/10.1109/INFVIS.2001.963279

Zhang TY, Wang XM, Li ZZ, et al., 2017. A survey of network anomaly visualization. *Sci China Inform Sci*, 60(12):121101. https://doi.org/10.1007/s11432-016-0428-2

Zhao J, Cao N, Wen Z, et al., 2014. #FluxFlow: visual analysis of anomalous information spreading on social media. *IEEE Trans Vis Comput Graph*, 20(12):1773-1782. https://doi.org/10.1109/TVCG.2014.2346922

Zhao J, Liu Z, Dontcheva M, et al., 2015. MatrixWave: visual comparison of event sequence data. Proc 33rd Annual ACM Conf on Human Factors in Computing Systems, p.259-268. https://doi.org/10.1145/2702123.2702419

Zhou DW, He JR, Candan KS, et al., 2015a. MUVIR: multi-view rare category detection. Proc 24th Int Joint Conf on Artificial Intelligence, p.4098-4104.

Zhou DW, Wang KY, Cao N, et al., 2015b. Rare category detection on time-evolving graphs. IEEE Int Conf on Data Mining, p.1135-1140. https://doi.org/10.1109/ICDM.2015.120

Zhou DW, Karthikeyan A, Wang KY, et al., 2017. Discovering rare categories from graph streams. *Data Min Knowl Discov*, 31(2):400-423. https://doi.org/10.1007/s10618-016-0478-6