NetEvViz: Extending NodeXL for Dynamic Network Visualization

Udayan Khurana  
Computer Science Dept.  
University of Maryland  
udayan@cs.umd.edu

Viet-An Nguyen  
Computer Science Dept.  
University of Maryland  
vietan@cs.umd.edu

Hsueh-Chien Cheng  
Computer Science Dept.  
University of Maryland  
cheng@cs.umd.edu

Xi (Stephen) Chen  
Computer Science Dept.  
University of Maryland  
xichen@cs.umd.edu

ABSTRACT

We present NetEvViz, a visualization tool to help analysis of dynamic and static network data. While there are a few tools for analyzing static networks, where edge or node addition or deletion does not feature, there are none publicly available that deal with dynamic networks that change with time. Our tool is a Microsoft Excel plug-in and extends the codebase of NodeXL, a popular network visualization software. The key features of this work are a) the ability of the user to specify and edit temporal annotations to the network components in an Excel sheet, b) Visually see the dynamics of the network with multiple graph metrics plotted over the time span of the graph, called Timeline, and c) Temporal exploration of the network layout using different colors through a dynamic Timeslider. The objective of the new features presented in this paper are to let the data analysts, computer scientists and others be able to visually see the dynamics or evolution in a network, in an interactive manner. We presented NetEvViz to four different users of NodeXL and got an overall positive response in terms of utility of such a tool and the presented features for visualization of dynamic networks.

INTRODUCTION

The visualization of network structure has gained much attention recently. The analysis of large scale complex network formed by social media has become an interesting topic with great potential in terms of research and business. Through a combination of computers and the great pattern recognition and cognition capability of human visual system, visualization has been considered to be one of the most powerful approaches to the analysis of networks.

Dynamic network is an extension of conventional network with temporal attributes attached to the vertex/edge. Since the structure of a temporal network evolves over time, interesting questions related to temporal attributes can be formed. However, with vertices/edges creation/removal, the analysis of a dynamic network can be challenging. For example, the removal of vertices and edges from the graph can be distracting.

Of all the tools ever been developed to perform network visualization, NodeXL is a highly recognized tool based on Microsoft Excel which supports flexible manipulation of the visualization. However, there exists only limited ways to analyze dynamic network in the latest version of NodeXL. In order to make better analysis possible and easy to access, more advanced approach to visualize dynamic network have to be included in the tool.

In this work, we build a system prototype to visualize the differences of dynamic network. The prototype is an extension of NodeXL. Two temporal attributes, start time and end time, represent the time each vertex/edge entered/leaved the network. An example of dynamic network is a social network where each vertex corresponds to a specific user and each edge corresponds to the certain relation between users. The start/end time of a vertex is the time when the user joined/leaved the social network by creating/deleting the profile. The start/end time of an edge can be used to represent the time when a be-friend-with relation is established/removed.

To help analyze the temporal network, we develop a prototype to visualize the difference of a temporal network at two timestamps. Given the two timestamps specified by the user, the vertices/edges are categorized into groups with respect to their temporal attributes. Each group is assigned a different color which enables user to recognize subtle patterns otherwise hard to be seen. A metric summary over time is provided in the prototype system to give an overview of the temporal network and help select interesting timestamps to begin with further analysis.

RELATED WORK

Dynamic Network Visualization

Visualization is an important and useful approach to analyze network data. Network visualization techniques have been proposed in various domain including social networks [3], biological networks [11], computer networks [1]. With the recent popularity and availability of temporal or longitudinal network data, there has been an increasing interest in developing visualization techniques for dynamic networks. In general, these visualization techniques use two popular approaches to represent the network data.

Submitted as the final report for CMSC 734 Information Visualization.
Local metrics refer to the topological properties of individual nodes in the network. These metrics include measures such as degree, closeness, betweenness, eigenvector centrality, and PageRank. In addition, clustering coefficient and assortativity coefficient are important local metrics that quantify how well nodes locally connect to each other.

Group metrics describe the topological properties of a group or community of nodes in the network. These metrics include modularity, group evolution metrics, and other community detection metrics. Group metrics are useful for understanding how different communities within the network interact and evolve over time.

Global metrics measure the overall structure and properties of the network as a whole. These metrics include measures such as the number of nodes, edges, and density, as well as network centrality measures such as degree centrality, betweenness centrality, and eigenvector centrality. Global metrics provide a comprehensive view of the network and are useful for understanding the network's overall structure and function.

The first approach uses the traditional nodes and links representation to visualize the network, which is intuitive and easy to understand. To add the temporal information, the network topology metrics can be aggregated over multiple nodes to achieve values or distributions at higher levels. This can be done by selecting and dragging nodes to manually change the layout that was computed algorithmically. The network can also be filtered based upon a numerical attribute associated with each edge. The tool also contains algorithms for calculating network metrics like degree centrality, betweenness centrality, clustering coefficient, eigenvector centrality, and PageRank.

Besides measurements for static networks at different points in time, a number of metrics have been proposed to explicitly quantify the change of network over time. NodeXL is introduced as a novel dynamic centrality metric which measures how well connected a node is over time. NodeXL is a free to use and open source tool for Network Analysis and Visualization. NodeXL provides a rich set of functionalities for the visual analytics of static networks. In the next section, we will consider more detail the features of NodeXL and why it is a suitable candidate for extension towards dynamic network visualization.
end time of a node are essential inputs to the tool for mapping the temporal dimension of data. We introduced two new columns - Start Time and End Time to the Vertices and Edges worksheets, which specify the time when a node or an edge joined the left and the time after which it was no longer a member of the network, respectively. In many networks, a node just joins the network and stays for ever, where as, in others it can leave the network or break a relationship at a point in time. Hence, we consider it mandatory to enter the Start Time for temporal analysis, but the absence of an end time is understood to be the current time by default. Note that the absence of a timestamp in any of the rows does not affect any existing functionality and hence backwards compatibility is guaranteed. Currently, we have put these two columns as the first two of the user defined columns tentatively, and would propose this to be a part of the main stream worksheet template for NodeXL. For this version of our tool, we have designed two main features - Timeline, to view the dynamics of the network using a change in graph metrics, and a Timeslider for Temporal Comparison. Accordingly, we created a tab in the NodeXL ribbon. These can be seen in figure 1. The details of the features follow in the next two subsections.

Figure 1. The Edges worksheet of NEtEvViz for Twitter dataset with Start and End Times

Network Timeline

In order to get a high level overview of the dynamics of a network, we decided to present the NodeXL user with a timeline of different graph metrics. The essential idea here is to let the user see the change in metrics over time and identify points of interest with respect to those metrics. Graph metrics are different from node metrics in the sense that they present a property of the whole graph instead of components of the graph individually. Examples of such metrics are over-all node count, overall edge count, graph density, average betweenness centrality, average clustering coefficient amongst others. We believe that for a user analyzing a new dataset, getting an overview at the highest level is the first important insight. In this project, we implemented the three most basic metrics for temporal graphs, namely, temporal node count, temporal edge count and temporal edge-node density. The user can compute these metrics and plots by clicking the Timeline button on the Temporal tab of the NodeXL Ribbon. The user should also select a value of timesteps, named Steps from the three given options - 5, 10 or 15, the first one being the default value. According to the number of steps selected, the tool calculates the values of metrics at the different timepoints. The selection of actual timepoints is a separate problem in itself. One way of going about it is selecting the points of maximum change reflected in the graph. However, time being a continuous quantity, it is a rather ill-defined problem to take such an approach. Another option could be to calculate and plot these metrics at all times where there was a change. However, there are obvious performance issues with this approach. In the absence of an obvious method to figure out the timepoints of importance, we adopted the selection of timepoints at uniform intervals of time. Starting from the start time of the first edge and ending in the last time of the edge, the timeline consists of 5,10 or 15 equally spaced timepoints. Figure 2 shows the timeline metric values and plots for a dataset consisting of Twitter replies and mentions described in the section on Evaluation.

Timeslider

NodeXL provides a set of dynamic filter sliders each of which corresponds to a column specifying a certain feature of the nodes or edges in the network. These double-headed sliders allow users to interactively filter out nodes and/or edges from the network by selecting the ranges of different features. In NetEvViz, we introduce Timeslider which is implemented based on the built-in dynamic filter sliders. Similar to the above-mentioned sliders, the Timeslider is also a double-headed trackbar which allows users to move its two handlers in order to select a time period. A major difference is that the Timeslider’s values are drawn from the contents of both the Start Time and the End Time columns. Thus, the range of the Timeslider would be from the earliest time stamp in the Start Time column to the latest time stamp in the End Time column. We denote these two time points $t_{\text{min}}$ and $t_{\text{max}}$ respectively.

By selecting two time points of interest, the users can visualize the differences of the network at the two specified time points. The two selected time points are denoted by $t_{\text{lower}}$ and $t_{\text{upper}}$ respectively. Figure 3 shows how the Lower and Upper Selected Time divide the whole time range into three periods. Each edges of the network can be categorized into one of the six disjoint sets based on the relationship between their start and end times with the lower and upper selected times.

Let $t_s$ and $t_e$ denote the start and end time of an edge. The six categories of edges are as follows:

1. $t_{\text{min}} \leq t_s < t_e < t_{\text{lower}}$: Set of edges that were added and deleted before the lower selected time $t_{\text{lower}}$.
2. $t_{\text{lower}} < t_s < t_e < t_{\text{upper}}$: Set of edges that were added and deleted between (but not inclusive) the lower selected...
Figure 2. A Timeline of ten steps, showing values for edge count, node count and edge:node density. The plot is for edge count.

Figure 3. The Timeslider with two selected time points dividing edges into six categories

1. Minimum Start Time $t_{\text{min}}$.
2. Lower Selected Time $t_{\text{lower}}$.
3. Upper Selected Time $t_{\text{upper}}$.
4. Maximum End Time $t_{\text{max}}$.

3. $t_{\text{upper}} < t_s < t_e \leq t_{\text{max}}$: Set of edges that were added and deleted after the upper selected time $t_{\text{max}}$.
4. $t_{\text{min}} \leq t_s < t_{\text{lower}} < t_e < t_{\text{upper}}$: Set of edges that were added before the lower selected time $t_{\text{min}}$ and deleted between (but not inclusive) the two selected time points.
5. $t_{\text{lower}} < t_s \leq t_{\text{upper}} < t_e \leq t_{\text{max}}$: Set of edges that were added between the two selected time points and deleted after the upper selected time $t_{\text{upper}}$.
6. $t_{\text{min}} \leq t_s \leq t_{\text{lower}} < t_{\text{upper}} \leq t_e \leq t_{\text{max}}$: Set of edges that were added before the lower selected time $t_{\text{lower}}$ and deleted after the upper selected time $t_{\text{upper}}$.

In order to help the user distinguish different sets of edges described above, we add four buttons to let the user choose the colors to represent four different types of edges. The four buttons are shown in Figure 4. In principle, since there are six categories of edges, one can use six colors. However, since edges in categories 1, 2 and 3 are in common in which they do not exist in the network at either of the two selected time points, we decide to use the same color to represent them. By default, this group of edges is coded in gray and denoted by “Not exist in both timestamps” as shown in Figure 4. The other three edge categories are represented by three different colors to be chosen by the user. The default color for category 4 “Only exist in the first timestamp” is green, for category 5 “Only exist in the second timestamp” is blue and for category 6 “Exist in both timestamps” is pink.

Figure 4. The Timeslider with double headers to select two time points of interest and four buttons to select colors for different edge categories

To illustrate, we use a toy example network containing 11 nodes and 21 edges. Each edge in the network is assigned artificial start and end time ranging from 14-Oct-2010 12:00 AM to 25-Oct-2010 12:00 AM. Figure 5 shows the visualization of the network using NodeXL. Figure 6 shows the visualization of the network with the same layout when the
Figure 5. Original network visualization by NodeXL.

Timeslider is used and the two time points selected are 14-Oct-2010 10:03 PM and 21-Oct-2010 12:38 AM respectively. In this new visualization, we can clearly see edges in different categories with respect to the two selected time points. The set of edges which only appear at the first time point is coded in green containing 6 following edges (Wen, Sameer), (Claudio, Wen), (Martin, Jill), (Jill, Steve), (Jill, Anushka) and (Steve, Anushka). There are only two edges which appear in both time points (Craig, Jill) and (Craig, Steve) which are coded in pink. The blue edges are the ones which only appear at the second time point. This set contains 6 edges (Alex, Anwar), (Alex, Sameer), (Wen, Kent), (Craig, Sameer), (Sameer, Kent) and (Jill, Kent). The remaining edges are coded in gray representing the set that does not exist at both time points.

EVALUATION
In order to assess the effectiveness of our new features added to NodeXL, we design and perform an experimental user study. We first decide a list of tasks related to temporal network analysis together with a questionnaire for the user to evaluate and make suggestions after performing the tasks. A pilot study is then conducted on one subject to evaluate our designed tasks. Based on the results and feedbacks from the pilot study, we modify the tasks and questionnaire accordingly and perform the actual usability test on X subjects.

Dataset
The dataset used in our evaluation is collected from Twitter using the NodeXL’s Twitter Search importer [3] focusing on the toss-up gubernatorial races in the 2010 United States midterm elections in Georgia between Nathan Deal and Roy Barnes. The importer collects all Twitter users who included a candidate’s full name in their Tweet (e.g., ”Nathan Deal”) or alternate variations of their names (e.g., ”NathanDeal”). We consider each Twitter user a node and a reply-to or mention relationship is an edge in our network. The choice of including only reply-to’s and mention’s but not the following relationships is because only reply-to and mention relationships have the exact time stamps. The data were collected on three days: a) Oct 12, three weeks before the election, b) Nov 2, on the election day and c) Nov 4, post election day. Totally there are 1188 edges and 1015 nodes in the network.

Since our visualization supports networks which have edge additions and deletions, we need the data for the edges’ End Time. Unfortunately, there is no notion of “deletion” of a reply-to or mention relationship. Thus, we artificially generate the End Times for all the edges by adding a random of days in the range of [3, 10] to the edges’ Start Time. A node’s timestamp is that of the first edge that the node has in our dataset.

Usability Study
Subjects
We have asked 5 graduate students, denoted by P1 to P5, from the Department of Computer Science, University of Maryland, as participants. The group includes 3 male and 2 female students. Participants should have enough knowledge in Excel and experience in NodeXL. The reason to recruit users with NodeXL experience is that participants will understand more the purpose and method of visualization the network data from spreadsheet so they will be more highly motivated in performing the tasks. We tried to ensure that by using questionnaire and recruiting people from CS department. Among them, P1, P3, P4 and P5 are familiar with using NodeXL. P1 and P3 are also familiar with the Twitter data.

Experimental Setup and Procedure
The evaluations were conducted on an Intel Core Duo 2 2.5Ghz laptop with 4GB of RAM. We used the Twitter dataset described in this section for the experiments. Participants are recruited from class of CMSC734. The reason is in this class there is a project using NodeXL so everyone is supposed
to have NodeXL experience. Subjects were tested independently, using the following procedure:

1. Before the experiment, participants will be explained in general what they are expected to perform in the experiment, read and sign the Experimental Consent Agreement.
2. Prior to the first session, participants completed a questionnaire focusing on their NodeXL and temporal network analysis experience.
3. Prior to the first analysis task, subjects were given a 10-min training by the experimenter including a walkthrough of NodeXL, introduction of the new features and instructions on completing the analysis tasks.
   1) Duration of the training (time) is 10 mins, including tutorial of NodeXL and NetEvVis, and two sample tasks.
   2) The new features explained are:
      a) Dynamic exploration of time using time slider
      b) Selecting two time points and contrast the difference visually
      c) Timeline of multiple graph metrics
      d) Directly editable data from the worksheet
   3) We also provide some sample tasks. A few sample tasks can benefit the participants to get familiar with the operation of the system and the property of the test tasks. However, too many sample tasks may leave prior assumptions to the participants and bias the main task. So we decide to ask the users to do the following sample tasks while introducing new feature of NetEvViz to them:
      a) Click the Timeline button to some generate graph metrics.
      b) Slide the time slider randomly to see if there is changes in the graph.
4. The participants were asked to perform the following tasks. They were asked to think aloud of what they were doing and report any problem, and the experimenter made notes at the same time.

   Task 1: Visualize network statistics over time (Note all time are in year 2010)
   a) What is the number of edges added to the network during period T1 after period T2?
   b) What is the number of nodes added to the network during period T1 after period T2?
   c) How does the density of the network at time t1 change after time t2?
   Task 2: Visualize the differences of network at different points in time
   a) Move the handlers of the time sliders in order to compare the network at time t1 and t2
   b) Change the colors of edges that appear in the network at time t1 but not at time t2 to Red
   c) Change the colors of edges that appear in the network at time t2 but not at time t1 to Blue
   d) Change the colors of edges that appear in the network at both t1 and t2 to Green
   e) How many edges appear in the network at time t1 but not at time t2?
   f) How many edges appear in the network at time t2 but not at time t1?
   g) How many edges appear in the network at both t1 and t2?

   Task 3: Visualize temporal network with interactive spreadsheet
   a) Add the following edges to the Edge Sheet
   A B t1 t2 (create 3-5 interesting edges to add to the sheet)
   b) List all the edges that appear in the network at time t1 but not time t2
   c) List all the edges that appear in the network at time t2 but not time t1
   d) List all the edges that appear in the network at both t1 and t2

5. Finally, all participants were asked to fill out a post-experiment Usability Questionnaire described below. We used 9-point Likert scale for measuring specific variables. They are also asked to leave comments for the experimental system.

   1) How to you rate your overall performance on the tasks given?
   2) I am comfortable with using the timeline.
   3) I find the timeline plots showing the three network statistics (number of edges, number of nodes and density) changing over time useful.
   4) I would suggest adding the following network statistics
   5) I am comfortable with using the time slider
   6) I find the feature of changing the color for different types of edges useful
   7) I am comfortable with detecting different types of edges given two selected time points
   8) I am comfortable with changing the color of different types of edges
   9) I find the ability to edit the spreadsheet while visualizing the network is useful
   10) I find the time slider functions smoothly and correctly as I move the handlers
   11) Are you aware of any other tool/software which allows you to compare network at two time points? If yes, please provide the name of the tool/software
   12) My overall impression of the NetEvViz is
   13) I think NetEvViz will be an useful extension for NodeXL
   14) Please leave any comments to the NetEvViz team. Any comment/feedback is much appreciated

Results and Analysis
As mention before, we have conducted the experiments with 5 participants independently. The main session of experiment is to perform 3 tasks with 3-7 subtasks each. In average, it took participants 1-3 minutes for each subtasks. The average time used is 6, 15 mins and 10min for task 1 to 3, respectively. The major part of the time is spent on reading and understanding the tasks, understand of the meaning of each metric generated by the timeline, and finding and selecting specific edges in the visualization.

P1 finished all the tasks successfully, but he had some troubles in understanding the function of time slider. He thought...
the time slider defines a time range, not only two time points. He said that in his opinion, I think you should use two single sliders instead of a range slider.

P2 performed the tasks quickly. He thinks that the concept of using time slider for temporal network analysis is novel. But he encountered some problem in date selection.

P3 did not finish task 3 because he was not able to select edges and nodes in the visualization. Since we are based on NodeXL source and have not changed any code in node/edge selection, we can assume this was an existing problem in NodeXL.

P4 did a careful study and provided many useful suggestions, esp. in color selection and coding. She mentioned that the labels should be attached to the color button. She also commented on the color coding of the visualization. She said it is very difficult to detect the edges that are activated during specific time period. So it would be much better to filter out gray edges. We have adopted her first suggestion but for the time being have not yet implemented the solution to reduce occlusion of target edges.

P5 also likes the idea of using color to compare the temporal network at different time point. She also thinks that it is interesting to use timeline to generate and compare metrics over time of the network. However, she thinks it would be more convenient to add the selection of date and time in the excel worksheet so that when adding/editing time of edges, users do not need to manually enter every time.

After the test, participants were asked to fill out and post-experiment usability questionnaire as described previously. The questions cover the overall performance, usability of the timeline and time slider, the visualization of the color coding, and the effectiveness of performing the tasks. The average score of usability is 6.38 out of the likert scale of 9, with a standard deviation of 2.23. Among the questions, users have most highly evaluation on question 6 I find the feature of changing the color for different types of edges useful (8.4 points out of 9); question 8 I am comfortable with changing the color of different types of edges (got 8.2 points out of 9) and question 12 My overall impression of the NetEvViz is positive (7.4/9). The lowest scored questions are question 10 I find the time slider functions smoothly and correctly as I move the handlers (4.6/9); question 1 How to you rate your overall performance on the tasks given(5.2/9) and question 5 I am comfortable with using the time slider (5.4/9).

From this questionnaire and the comments of participants, we can conclude that users have a positive experience in using the NetEvViz overall. They found it useful to use NetEvViz to visualize and compare temporal network at two different time points. They are especially in favor of the feature of color selection and comparison. However, they also report the occlusion of edges in clutter made it hard to detect target edges, and inconvenience in selecting date/time in time slider and selecting edge/node in the visualization. Such problems may be small, and may be a problem inherited from Excel and NodeXL which we are unable to fix, but it can lead to the reduction of effectiveness to perform tasks.

CONCLUSION
In this work, a prototype is built upon NodeXL to explore the potential of using visualization to help the analysis of dynamic network. The prototype contains two new features, Timeline and Timeslider. Timeline gives the user a summary of the network over time, which allows the user to obtain an overview of the changes in the network. Timeslider allows for precise comparison of the differences in the network, which are highlighted with colors, given two timestamps which are specified by the user.

A user study was conducted to confirm the usefulness of our prototype. 5 participants with NodeXL experience were recruited. The study follows the procedure of training, pre-experiment questionnaire, performing predefined tasks and post-experiment usability questionnaire. Over all, from the feedback of the user study, we can summarize that users have a positive experience and thinks it is useful to use our NetEvViz features to visualize and analysis temporal network data jointly with Excel. Users consider the system is novel in the idea of using time slider to select to time point and then compare the network in different color in the same visualization. The users also like the feature of graph metrics over time and editing edges and nodes in the spreadsheet dynamically. However, they also think some inconvenience in user interface and color coding affect their efficiency in performing the tasks.

The prototype built in this work is the first step to understand how NodeXL can be used to analyze dynamic network. Some possible future work include:

- More sophisticated metrics can be added to the Timeline beside the three trivial metrics we implemented.
- The Timeslider can be extended to color the vertices in a similar way we color edges. Also, it would be interesting to look at visual properties beside color (e.g. shape or opacity).
- Support the analysis of egocentric network by allowing the users to specify the vertex they are interested in.

Proposals to the NodeXL team
Based on our experience with the development of this tool to visualize dynamic networks and the due to the benefits of being able to visualize the evolution or change in a network, we would like to propose the NodeXL development team to incorporate the following features:

1. Temporal annotation columns, i.e. Start Time and End Time should be first class citizens of the Network analysis world.

2. Time based filtering of nodes and edges should be present as a separate feature, for instance, using a double slider as shown earlier in this paper.
3. A user should be able to view two versions of a network at the same time, i.e. side by side of each other. The user should be able to switch back and forth between a single and double layout view mode.

4. Network, node and edge metrics should be extended to cover temporal change.

CONTRIBUTIONS AND ACKNOWLEDGEMENTS
Here are the contributions of the four team members:

1. Udayan Khurana contributed with ideas and discussions about the subject, exploring NodeXL codebase, designed the Timeline feature, ribbon changes, made the video, and contributed in writing.

2. Viet-An Nguyen contributed with ideas and discussions about the subject, exploring related work, designed the Timeslider feature, processed the data and contributed in designing the user study and writing.

3. Hsueh-Chien Cheng contributed with ideas and discussions about the subject, designed the Timeslider feature, conducted user study and contributed in writing.

4. Stephen Xi Chen contributed to the design of the user study, conducted user study and contributed in writing.

We would like to acknowledge Dr. Jae-wook Ahn (University of Maryland) who mentored the team, providing his guidance and expertise in the subject matters. We also thank Prof. Ben Shneiderman (University of Maryland) and Mr. Tony Capone (Microsoft) for the useful discussions and suggestions on related to ideas and implementation on this project. Finally, we thank the anonymous subjects of the usability testing for their time and feedback.

ADDITIONAL AUTHORS

REFERENCES


