Eurographics / IEEE Symposium on Visualization 2011 (EuroVis 2011)
H. Hauser, H. Pfister, and J. J. van Wijk
(Guest Editors)

Volume 30 (2011), Number 3

Visual Recommendations for Network Navigation

Tarik Crnovrsanin[1], Isaac Liao[1], Yingcai Wu[†1], Kwan-Liu Ma[1]

[1]The Visualization and Interface Design Innovation (VIDi) Research Group
University of California, Davis

Abstract

Understanding large, complex networks is important for many critical tasks, including decision making, process optimization, and threat detection. Existing network analysis tools often lack intuitive interfaces to support the exploration of large scale data. We present a visual recommendation system to help guide users during navigation of network data. Collaborative filtering, similarity metrics, and relative importance are used to generate recommendations of potentially significant nodes for users to explore. In addition, graph layout and node visibility are adjusted in real-time to accommodate recommendation display and to reduce visual clutter. Case studies are presented to show how our design can improve network exploration.

Categories and Subject Descriptors (according to ACM CCS): [Visual Knowledge Discovery]: Data Filtering—Graph/Network Data—Human-Computer Interaction

1. INTRODUCTION

Network analysis finds many applications in biology, computer science, economics, and sociology. Visual inspection of a network is an effective way to begin an analysis. However, when a network is large or complex, finding an optimal layout for conveying the network's essential structure becomes a challenge [TLM05]. Network visualization research has largely focused on creating fast, aesthetic, hierarchical graph layouts [Fur86] to facilitate network exploration. However, these layouts often require specialized domain knowledge and analysis skills to successfully navigate large or complex networks. To allow users to explore networks with no prior knowledge, it would be desirable for systems to show only the most relevant portions of networks to them, while suggesting directions for potential exploration. Such recommendation systems have been constructed for other navigation and information-seeking applications [KSC*08, LSY03, GW09].

We present a network visualization and exploration system which recommends relevant nodes to users based on relative importance, similarity, and past user interaction. Additionally, we present an accompanying layout method that reduces visual clutter and edge crossings by considering which nodes are more likely to appear in subsequent navigation steps. Example applications of our system are illustrated using three real-world datasets. We find that our system enables the discovery of hidden, interesting structures in complex network data.

2. RELATED WORK

2.1. Intelligent Visualization

Researchers have designed a variety of intelligent techniques to improve the effectiveness of visualization systems. In the field of intelligent user interfaces, systems can generally be divided into task-based [Cas91], data property-based [Mac86, RKMG94], or hybrid systems [ZC03]. To clarify, ours is a hybrid system which takes into account both data properties and user interaction history to generate recommendations.

More recently, researchers have focused on modeling user context and behavior to assist visualization tasks. For instance, a smart visual dialog system has been developed to help users explore large, complex data sets [WZ08, ZHP*06]. This system automatically generates tailored visualizations to meet users' requirements based on their navigation contexts. Gotz and Wen [GW09] introduced a system which infers a user's intended visual task based on their behavior, and automatically recommends alternative visualizations to them. Finally, Gretarsson et al. [GOB*10] presented an interactive graph-based interface for users to build

† Corresponding Author

and refine their item-preference profiles in Facebook, based on collaborative analysis of other users' preferences. In the context of these previous works, our system produces a visualization based on not only the current user's behavior and data attributes, but also those of all other users who have used the system in the past. Thus, our system can recommend either the most-explored or the least-explored areas of data to users.

In scientific visualization, researchers often employ machine learning techniques to facilitate visual analysis tasks, such as data classification [TLM05] and pattern detection [TM05]. Intelligent algorithms have also been utilized to improve the intuitiveness of visual data exploration by recommending alternative visualizations [KSC*08]. In information visualization, researchers advocate visual data mining, which integrates visualization techniques into data mining processes [Kei02]. Carenini et al. [CR09] introduced a multimedia interface that combines information visualization and opinion mining techniques for the visual analysis of opinions across multiple entities. In contrast to existing work, our system aims to produce intelligent recommendations for network navigation as well as on-the-fly optimizations of graph layout to accommodate recommendation display, which we believe to be a combination of techniques that has rarely been explored before.

2.2. Large Network Visualization

Large network visualization is an important topic in the field of information visualization (for review, please refer to [vLKS*10]). Researchers often use node-link diagrams [CZQ*07], treemap layouts [Shn92], and/or matrix views [EDGH08] to visually represent network data. Our work builds upon familiar node-link diagrams to facilitate visual search, analysis, and exploration of large network connectivity.

Network visualization techniques can generally be classified as either top-down or bottom-up techniques. Top-down approaches start with an overview of the entire network and focus on regions of interest by filtering and zooming. Shen et al. [SMER06] employed semantics, structural abstraction methods, and graph ontology to prune large networks. Perer and Shneiderman [PS08] designed a user interface infrastructure that allows for maintaining user interaction history, measuring visualization progress, and keeping users informed. This infrastructure can be used in large network visualization. Fast graph layout algorithms [HJ07], clustering methods [BM03], or multi-scale clustering techniques [vHvW05] are often employed to create effective network overviews. On the other hand, these approaches may not be very useful in applications where users only need to explore and analyze a small part of the data.

To address this issue, researchers have introduced several bottom-up techniques. These approaches start from a single selected node and its immediate context. Additional relevant nodes and connections are revealed only on demand, based on graph structure or specialized degree-of-interest functions. Moscovich et al. [MCH*09] designed two intuitive interaction techniques called "Link Sliding" and "Bring & Go" for navigating large networks. Heer and Boyd [HB05] presented a visualization method which only shows a focus node's neighboring nodes up to a certain level. Similarly, Elmqvist and Fekete [EF09] described a bottom-up system based on hierarchy traversal methods, including above traversal, below traversal, and level traversal. These methods are useful when the inherent graph structure is more important than other properties for the task at hand. For other applications, where node/edge attributes are the focus of analysis, researchers create specialized degree-of-interest (DOI) functions. Furnas [Fur86] introduced a DOI function to evaluate the importance of a selected node based on distance and a priori interest. Van Ham and Perer [vHP09] extended this function to operate on embedded attributes and graph topology, as well as user-generated search actions. Their system can then suggest nodes with the highest degrees of interest for users to explore. Our system differs from this work in several important ways. First, in addition to matching nodes based on attributes of the current selection (which we call relevance filtering), we also take prior user interaction history and relative importance into account. Furthermore, our system's interaction history includes not only the initial search node, but also how the user proceeded from there. Finally, our system uses a custom layout to display recommendation results. This layout is adjusted dynamically during the visualization process to accommodate recommendation display and to reduce visual clutter.

Our technique provides a top-down overview, but is mostly a bottom-up approach. To generate recommendations, our system uses eigenvector centrality, a robust importance metric that is at the core of ranking algorithms such as Google's PageRank [BP98] and Hyperlink-Induced Topic Search (HITS; [Kle99]). Since navigation recommendations are based on both data attributes and prior user interaction history, our approach is most likely to be useful in providing assistance to users navigating unfamiliar data.

3. SYSTEM DESIGN

Figure 1 gives a broad overview of our system's processing steps. User interaction consists mainly of sequential selections of nodes to explore. When a node is selected, the recommendation system uses collaborative filtering and relevance filtering to recommend related nodes. These recommendations are displayed using a suggestion-aware layout that we have developed specifically for this application. User interaction history is saved for future input to collaborative filtering.

Our system is built on top of Netzen, which is a software tool for the analysis and visualization of social net-

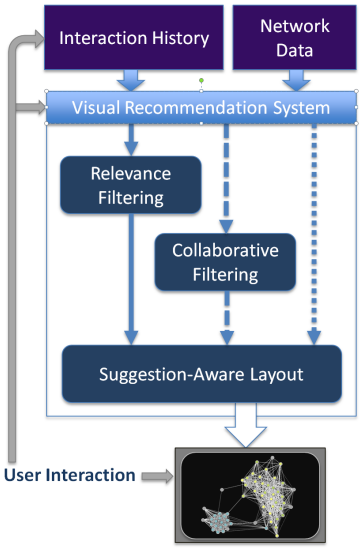


Figure 1: *Overview of system workflow. The user selects a node and the interaction is logged. Collaborative filtering, relevance filtering, or a combination of both approaches are used to generate recommendations. Finally, recommendations are passed to the suggestion-aware layout algorithm and displayed to the user.*

works (http://vis.cs.ucdavis.edu/~correac/netzen/). Netzen provides several layouts, centrality metrics, and filtering options.

3.1. Recommendation System

Our recommendation system is a hybrid of two approaches: collaborative filtering and relevance filtering.

Item-to-item collaborative filtering [LSY03] is the algorithm used by Amazon.com to recommend items to customers based on the item they are currently viewing. When users interact with our system, a vector of user selections is recorded. At run time, the application loads all past user history into a table with rows of user interaction vectors and n columns of nodes. When a user selects a node, we calculate the cosine similarity between the selected node and every other node:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

where A is a vector representing the selected node in higher dimensional space, B is a vector for each other node, and θ is the angle between them. Taking the cosine of this angle gives a similarity value ranging from 0 to 1, which is called the cosine similarity. Thus, we have a cosine similarity between the current node and every other node (if there are n nodes, then we calculate $n-1$ cosines). Nodes with the greatest cosine similarity to the selected node tend to be selected by users after selecting the current node. Therefore, cosine similarity can be used as an estimated conditional probability that the user will click on a particular node next, given their currently selected node, and is used by our system to find candidate nodes for recommendation.

Relevance filtering ranks nodes based on similarity and relative importance. The motivation for relevance filtering is as follows: If the user is interested in certain node properties, they should be shown nodes that are similar to the currently selected node with regards to these properties. In addition, users should be shown nodes that are relatively important to the currently selected node.

First, if the user has selected properties of interest using the ontology graph of node properties (see User Interface section), we find other nodes that are similar to the currently selected node with respect to the chosen properties. A similarity value is assigned to each other node by calculating the Euclidean distance for chosen properties between the currently selected node and each other node.

Second, the relative importance of potential recommendations is calculated using eigenvector centrality sensitivity [CCM10]. Eigenvector centrality is a measure of the importance of a node in a network, and is used by the PageRank [BP98] and Hyperlink-Induced Topic Search [Kle99] algorithms. Unlike other centrality metrics, which base importance on how many connections a node has, eigenvector centrality also weights connections based on the importance of linked nodes: a single connection to a highly important node carries more weight than many connections to nodes of low importance. Eigenvector centrality sensitivity extends this notion to derive the importance of nodes relative to each other. While centrality gives one value per node, sensitivity gives a value for every possible pair of nodes in a network. To calculate a reference node's sensitivity to a target node, the reference node's initial centrality is calculated, each edge of the target node is removed one at a time, and the centrality of the reference node is recalculated after each removal. The negative changes in centrality of the reference node give a measure of how important the target node is to the reference node. For instance, if removing a target node's edges resulted in large decreases in the reference node's centrality, then the reference node is said to be highly sensitive to the target node - that is, the target node has high relative importance to the reference node. This can be summarized in the following, from [CCM10]:

$$\frac{\partial x}{\partial t_i} = -Q^{+} \frac{\partial Q}{\partial t_i} x$$

where x is eigenvector centrality, t_i is a degree variable, Q is the subtraction of the identity matrix from the adjacency matrix of the network ($Q = A - I$), and Q^{+} is the pseudo-inverse of Q.

Similarity and relative importance are multiplied together

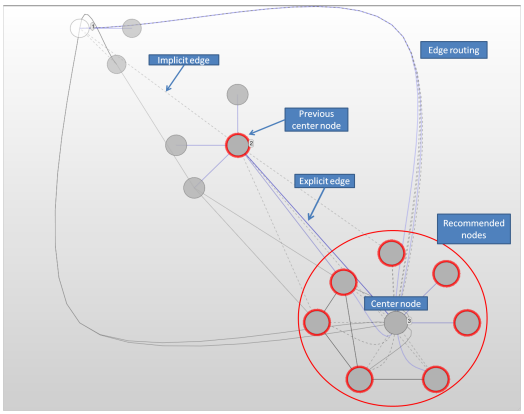


Figure 2: *Components of our network visualization. When a node is selected, up to eight recommended nodes (red rings) are displayed in a circle around it. Recommendations are based on interaction history across the current user and all previous users (collaborative filtering), as well as centrality and similarity metrics. Explicit edges are direct connections that exist in the network, while implicit edges are indirect connections. Note that for an implicitly linked node to be recommended, it must be in the top eight most relevant nodes for the current selection.*

to provide values for relevance filtering. Finally, recommendations are generated using a weighted average of collaborative filtering and relevance filtering; if more user interaction history is available, collaborative filtering is given a greater weight. Weights are adjusted as the system collects more user interaction history. Thus, our hybrid system is able to provide recommendations without any prior user interaction, and shifts to a more history-influenced approach as more information becomes available.

3.2. Suggestion-Aware Layout

Laying out a graph to accommodate visual recommendations is non-trivial. First, displaying recommendations using a force-directed layout [FER91] leads to excessive visual clutter if nodes are kept immobile. On the other hand, allowing nodes to move while the entire layout is recalculated after each selection step causes users to lose their navigation context. Second, if users select nodes that are part of previous navigation steps, keeping the current navigation path displayed contributes to visual clutter. Third, displaying duplicates of nodes that are recommended at different navigation steps causes confusion. Therefore, we deemed it necessary to design a new layout algorithm, which we call a suggestion-aware layout, to support intuitive network exploration supported by context-sensitive node recommendations.

As nodes are sequentially selected, the selection history forms a path traversing the network. When there are n selections in the interaction history, this path is displayed as a logarithmic curve, $y = \log|x| * c$, where x and y are Cartesian coordinates, c is a scaling factor, and $x \in \{-1,-2,-3,\ldots,-n\}$, with the most recently selected node corresponding to $x = -1$. We decided to use a logarithmic curve for several reasons. First, the shape of the logarithmic curve helps prevent edges from bisecting nodes that they are not connected to. Second, since current and recently selected nodes are more relevant to the user, the spacing of the logarithmic curve ensures that they are allocated more screen space than older nodes. In other words, previously selected nodes are plotted along a logarithmic curve that arcs up and to the left; new nodes appear in the bottom right, while older nodes fade, shrink, and eventually disappear in the top left. As more nodes are selected, the logarithmic curve provides the illusion of the navigation path diminishing into the horizon. To prevent visual clutter, selecting a node from a previous navigation step causes the layout to reset to that step, hiding all nodes that were selected afterwards. We chose to reflect the logarithmic curve about the y-axis because we believed that it was more intuitive for older nodes to be placed on the left, since timelines conventionally proceed from left to right. Thus, the currently selected node is emphasized both by always being located in the bottom right, and by being allocated the most screen space.

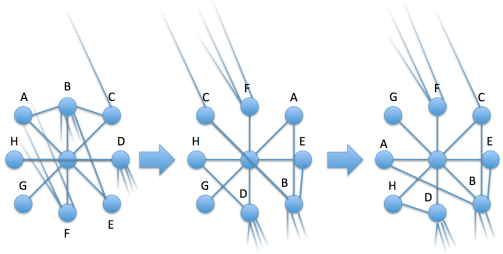


Figure 3: *The suggestion-aware layout first calculates node positions based on outer edge connections, then improves node positions by looking at inner edge connections. The layout minimizes edge crossings and edge-node bisections (middle panel).*

Our system displays three different types of edges: implicit, explicit, and normal edges (Figure 2). Implicit and explicit edges are links to recommended nodes. However, explicit edges (solid blue lines) exist in the dataset, while implicit edges (dashed lines) do not. The existence of an implicit edge signifies a strong correlation between two nodes, even though there is no direct connection in the dataset. Thus, it is important to show both explicit and implicit relations [DCW11]. Normal edges (solid black lines) exist in the dataset, but connect nodes that weren't directly recommended from each other. Normal edges provide structural information about the data. Longer edges, which connect

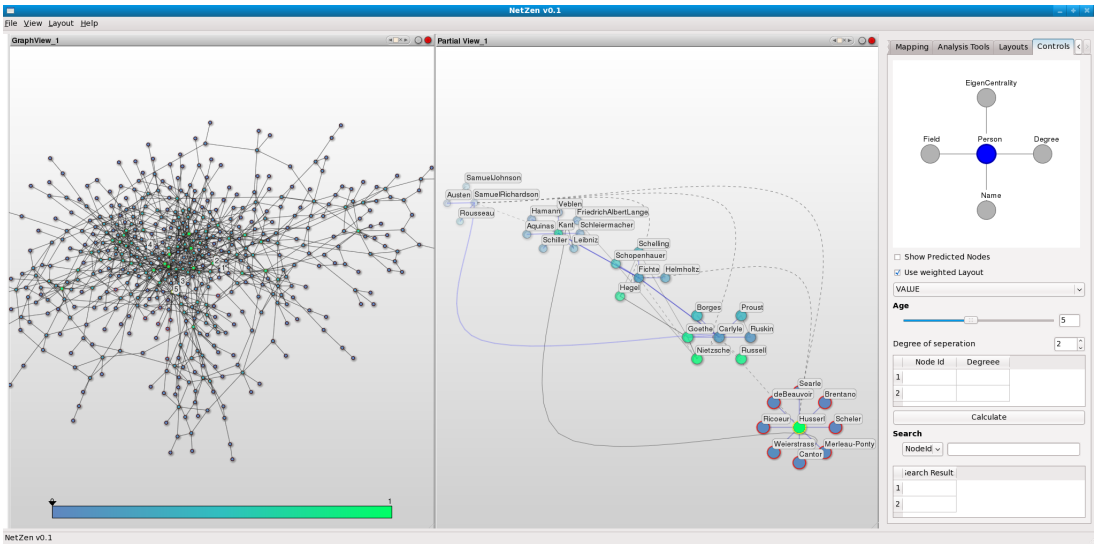


Figure 4: *The user interface. The left panel is an overview of the graph. The center panel is our recommendation system with suggestion-aware layout. An ontology graph of all node properties is in the upper right. The bottom right panel allows users to restrict displayed nodes to specified search terms.*

nodes that are more than one step apart, are routed around the outside of the main navigation path using splines. To allow users to easily differentiate between derived and actual data, implicit edges are routed towards the top right, while explicit and normal edges are routed towards the bottom left.

Arrangement of recommended nodes presents its own challenges. Recommendations should be placed to minimize edge crossings, edge-node bisections, and visual clutter (Figure 3). Initially, recommended nodes are arranged in a circle around the selected node in sequential order. Then, our system tries every permutation of node positions, picking the circular arrangement which produces the smallest force [FER91]. In the force calculation step, our system takes into account both connections to previously displayed nodes and to possible future recommendations (nodes that will appear if recommended nodes are clicked). Thus, recommended nodes tend to be placed near the top left of the selected node if they have more connections with already-displayed nodes, or near the bottom right of the selected node if they have more potential connections with future recommendations. Finally, a second pass is performed based on inner connections (between current recommendations and the currently selected node). Each node is only allowed to swap positions with its neighboring nodes, and again the circular arrangement resulting in the smallest net force is picked. Although the computational complexity of calculating forces between all nodes is high, the small number of nodes involved makes this negligible.

3.3. User Interface

The user interface is comprised of two network views and an ontology graph (Figure 4). It possible to create multiple views, but by default there is only one of each. The leftmost panel provides an overview of the network using a force-directed layout [FER91], which produces intuitive and aesthetically pleasing results. The high run time can be ameliorated by saving layout results for future use. By default, nodes are colored by eigenvector centrality [New07], with higher centrality corresponding to brighter color. However, users can map node color to any desired node property.

The large center panel shows our recommendation system with suggestion-aware layout. When a node is selected, it moves towards the bottom right of the screen, and recommendations for that node are shown in a circle around it. Additionally, if already-displayed nodes are recommended for the current selection, they are moved closer to the selected node. This allows users to more easily recognize currently relevant nodes, while avoiding confusion caused by displaying duplicate nodes. The overview and recommendation views are linked, and when a node is selected, red rings appear around recommended nodes in both views. Also, to facilitate comparison, nodes are numbered identically in both of the views. The maximum number of recommended nodes is user-adjustable. In this figure, five nodes have been selected so far, and the navigation history recedes towards the top left. To reduce visual clutter, selecting a node from a prior step causes the display to reset to that step. User selections are saved to be used as input for the recommendation system.

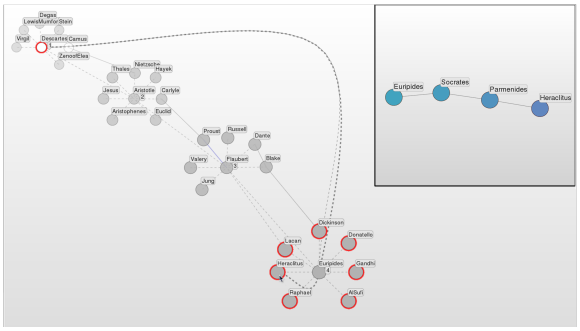


Figure 5: *The implicit link viewer shows how the implicit link between Euripides and Heraclitus is formed by displaying a path in the upper right corner.*

To make implicit links more comprehensible, we have implemented an implicit link viewer. When invoked by right-clicking on an implicit link, a pop-up view appears showing the path of explicit links composing an implicit link. This path is calculated using a breadth-first search, weighted by eigenvector centrality sensitivities.

The ontology graph in the upper right provides control over the recommendation system. Users select properties that they are interested in, and the recommendation system uses selected properties as input for relevance filtering. The ontology graph holds both raw and derived properties, and newly calculated data is added to the ontology graph in real time.

Finally, the search panel in the bottom right allows users to display nodes whose properties match specified search terms. This is particularly useful when the dataset is too large to visualize in the overview. Furthermore, if a user is investigating a particular subset of the network, they can restrict displayed nodes to those matching their search criteria.

4. CASE STUDIES

To illustrate the effectiveness of our system, we present three datasets as examples: the Genealogy of Influence network, a criminal organization social network, and a network of political blogs.

4.1. Genealogy of Influence

The Genealogy of Influence dataset is a network that describes how renowned intellectuals influenced each other's work throughout history. It was compiled by Mike Love [Lov] using Wikipedia to create associations. Nodes represent people, including artists, writers, mathematicians, philosophers, and scientists. Edges represent perceived intellectual influences, e.g., mentoring another person, building on another's work, or subscribing to similar schools of thought. The network was manually curated starting in 2005, but has since been uploaded to the data collaboration website Freebase (`http://www.freebase.com`), which allowed the community to add thousands of people and connections.

To show that our recommendation system creates implicit links that signify meaningful relationships between nodes, we find an implicit link to investigate. We select a person at random: Euripides, a Greek playwright who lived from approximately 480 to 406 BC. Figure 5 shows that all eight recommended nodes are implicitly linked to our current selection. In other words, our system believes that these 8 people, who are not directly connected to Euripides, are more important to our exploration of the network than anyone who is directly connected to him. To further investigate, we focus on one of the implicit links: the one between Euripides and one of the recommended nodes, Heraclitus. We select the implicit link, and a small view appears showing the shortest path of explicit links connecting the two.

The path shown is composed of Euripides, Socrates, Parmenides, and Heraclitus. Referring to Wikipedia, we find that Euripides was exposed to Socrates's work as a child, and that Socrates and Parmenides are portrayed discussing their ideas in the Dialogues of Plato. The final link between Parmenides and Heraclitus is the most intriguing since we cannot find out who influenced who. We do know that they were both ancient Greek philosophers who lived at the same time, and were therefore likely to have influenced each other's ideas. Thus, Parmenides and Heraclitus shaped the views of Socrates, who in turn shaped the views of Euripides. Using our implicit link viewer, we were able to discern how Euripides was likely indirectly influenced by Heraclitus.

In this example, implicit links represent indirect intellectual influence. It is important to keep in mind that the meaning of implicit links is dependent on the dataset and the properties that the user has selected in the ontology panel. Therefore, it is possible to generate implicit links not just for influence, but for any combination of properties in a dataset.

4.2. Criminal Organization Social Network

The VAST 2008 Grand Challenge provided a synthetic dataset [IEE] detailing the communications of a fictional criminal organization called the Paraiso Movement, led by a person named Ferdinando Catalano. It consists of a record of calls made between 400 cell phones over a 10-day period. One of the goals of the challenge was to characterize the structure of Ferdinando Catalano's social network.

We are told that Fernando Catalano calls his brother Estaban and his associates, the three Vidro brothers, most frequently. Additionally, David Vidro is believed to coordinate high-level Paraiso communications. However, false names were used to register the cell phones, so they are given numeric identifiers. We are told only that Ferdinando Catalano

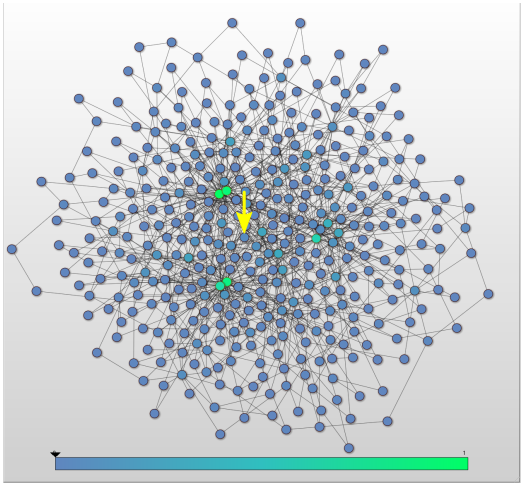


Figure 6: *Overview of a fictional criminal organization phone call network. Nodes represent cell phones and edges represent calls made between them. Nodes are colored by centrality (green: high centrality; blue: low centrality). Aside from several nodes with high centrality, it is difficult to distinguish any clusters or patterns in the network. The organization leader's cell phone (arrow) is indistinguishable from other cell phones.*

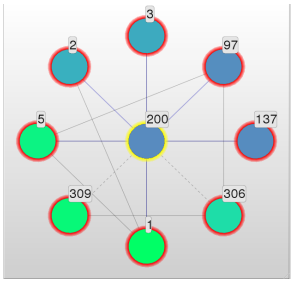
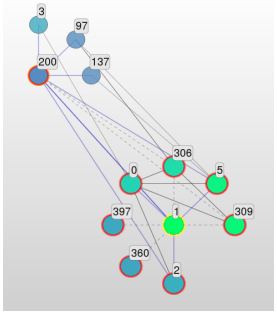


Figure 7: *Device #200 has both implicit and explicit connections with its recommendations.*

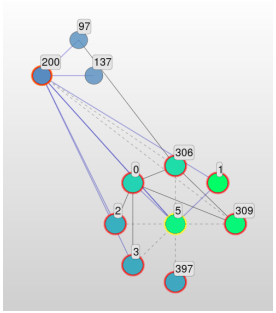
probably used cell phone #200; all other device-user associations must be derived.

It is logical to begin with an overview of the entire network (Figure 6). However, aside from several nodes with high centrality (colored in green), it is difficult to discern any significant features in the graph. Ferdinando's cell phone #200 (marked with an arrow) appears to be indistinguishable from most other devices. It is clear that an overview alone will not provide adequate information about the structure of Ferdinando's social network.

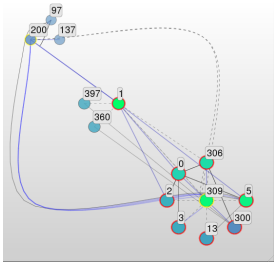
Instead, we decide to try a bottom-up approach, and select Ferdinando's cell phone #200, yielding the view shown in Figure 7. Out of the 8 recommended nodes (red rings), we see that there are explicit links (solid lines) with cell phones



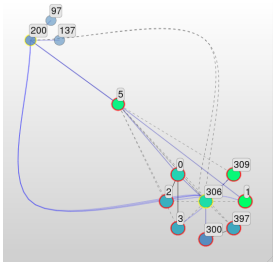
(a) Navigation from 200 to 1.



(b) Navigation from 200 to 5.



(c) From 200 to 1 to 309.



(d) From 200 to 5 to 306.

Figure 8: *(a) Navigating from 200 to 1 and (b) from 200 to 5 show similar recommendations of high centrality (306, 309). (c, d) Investigating 306 and 309 reveals only one new recommendation common to both, phone #300.*

1, 2, 3, and 5, meaning that there were actually calls placed between Ferdinando's cell phone and these phones. Also, there are implicit links (dashed lines) with cell phones 306 and 309, meaning that there were no direct calls between Ferdinando's cell phone and these phones, but our system has determined that they are relevant to cell phone #200. Furthermore, we can tell that cell phones 1, 5, 306, and 309 have very high centrality (green color). It is likely that cell phones 1 or 5 belong to David Vidro, because 1) they have high centrality, which we would expect if they are being used to coordinate the communication network, and 2) they are explicitly linked to cell phone #200, which we would expect if Ferdinando Catalano calls them frequently.

At this point, we are reasonably certain that cell phones 1 or 5 belong to David Vidro, but we don't know which one. Navigating to cell phone 1 yields the view shown in Figure 8(a), while navigating to cell phone 5 yields the view shown in Figure 8(b). In both cases, we notice that many of the same nodes are recommended: 0, 2, 306, 309, and 397. However, the types of links are different; 306 and 309 are both implicitly linked to 1 and 5, and explicitly linked to each other. This is especially interesting given the high centralities of 1, 5, 306, and 309: why would phones that call so many other members of the network not call each other?

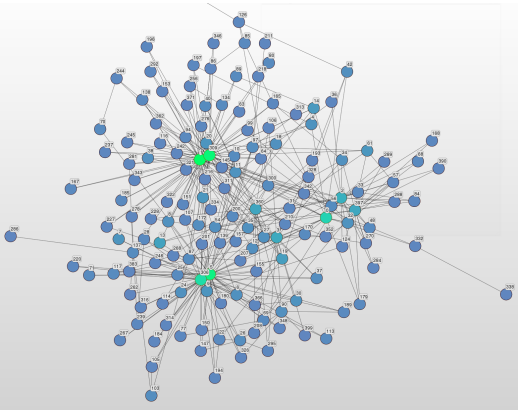


Figure 9: *Overview filtered to show only nodes that connect to group 1 (0, 1, 2, 3, 5, 200) and group 2 (300, 306, 309, 360, 397).*

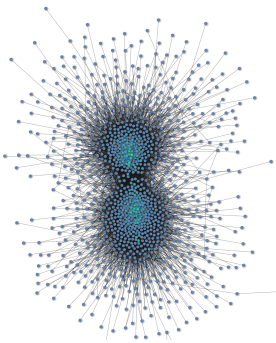


Figure 10: *Overview of Political Blogosphere dataset. Liberal blogs cluster together above and conservative blogs cluster together below.*

Based on this information, we decide to investigate phones 306 and 309. Selecting 306 yields the view shown in Figure 8(c), while selecting 309 yields the view shown in Figure 8(d). Interestingly, almost all of the same recommendations are carried over from 1 and 5 to 306 and 309. This tells us that there is a local cluster, and that we have already seen most of the nodes that are involved. The only new addition common to both is 300, which also has links to many of the nodes we've seen so far.

Further examination reveals that there are actually two clusters here: 0, 1, 2, 3, 5, and 200 (group 1) have explicit links to each other, while 300, 306, 309, 360 and 397 (group 2) have explicit links to each other. While we know that these two groups are somehow related, since they are implicitly cross-linked together, we don't know exactly how. Therefore, we try filtering our overview to show only nodes that link to members of these groups, yielding Figure 9. Interestingly, we notice that the force-directed layout in the overview panel has placed individual nodes from one group next to nodes from the other group, in 5 pairs. For instance, nodes 1 and 309 are next to each other, because phones 1 and 309 called the same individuals. The same holds true for 2-397, 3-360, and 5-306. This raises an interesting question: why would pairs of phones from these two groups call the same people, but not each other? Looking back in the dataset, we notice that phones from group 1 only made calls during days 1-7, while phones from group 2 only made calls during days 8-10. A possible explanation emerges: these two groups represent the same people, but they switched phones (from group 1 to group 2) at the end of day 7. Following the same logic, we find that Ferdinando Catalano most likely switched phones from #200 to #300 on day 8.

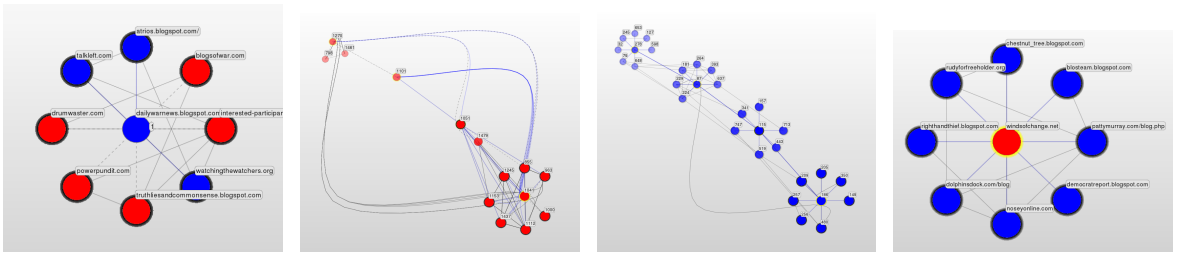
In summary, our recommendation system helped us discover a hidden relationship between two subnetworks in the dataset. This would not have been possible using an unfiltered network visualization.

4.3. Political Blogs

The Political Blogosphere dataset [AG05] is a snapshot of over 1,000 political blogs taken in February 2005. Nodes represent blogs, which are classified as either liberal or conservative. Edges represent instances where one blog cited another blog by posting a URL linking to them. As might be expected, liberal blogs tend to link to other liberal blogs, while conservative blogs tend to link to other conservative blogs. In our overview, this leads to a clear separation between the two groups, as shown in Figure 10.

Although there is a clear division between liberal and conservative blogs, our system allows us to find blogs that bridge the gap between them. Figure 11(a) shows one such example. Although a liberal (blue) blog is selected, more conservative (red) blogs are recommended than liberal blogs. In addition, closer inspection reveals that links to conservative blogs are implicit - that is, they do not actually exist in the dataset and are derived by our recommendation system. Thus, our system is able to reveal hidden, indirect connections that are not readily apparent.

It is also impossible to tell from Figure 10 whether there is any systematic difference in the way that blogs link to other blogs within the same party. However, using our system's search function to show only liberal blogs or only conservative blogs reveals an interesting difference: when navigating conservative blogs only (Figure 11(b)), successive selections seem to repeat recommendations (note how the circle of recommendations has moved with the selection towards the bottom right, leaving the middle area sparse). In contrast, when navigating liberal blogs only (Figure 11(c)), successive selections tend to have their own unique recommendations (note the circles of recommendations for previous selections). From this, we can conclude that the conservative blog network is more highly centralized, since the same



(a) "Bridge" blog. (b) Conservative blogs. (c) Liberal blogs. (d) Misclassified blog.

Figure 11: *(a) Despite division in the blog network across party lines, selecting dailywarnews.blogspot.com shows recommendations for both liberal (blue) and conservative (red) blogs. (b) Navigation of conservative blogs shows many repeat recommendations, while (c) navigation of liberal blogs shows many unique recommendations, indicating that conservative blogs are more centralized than liberal blogs. (d) When windsofchange.net is selected, only conservative blogs are recommended. This may indicate that it was misclassified as a liberal blog.*

set of important blogs are recommended repeatedly during navigation. In fact, referring to the original paper [AG05] reveals that the authors eventually came to the same conclusion through exploration. Thus, our system is able to reveal hidden structure that was masked by the sheer amount of data in the original overview.

Finally, blogs that are classified as liberal might cite more conservative blogs than liberal blogs, or vice versa. One example of this is shown in Figure 11(d). There are two possible explanations for this: either the blog was misclassified in the dataset, or the blog prefers to cite blogs from the opposing party (for instance, to attack statements made by opposed blogs). Upon reading the blog in this example, it becomes apparent that it was misclassified as conservative when its stance is highly liberal. Thus, our recommendation system may be used to identify abnormalities in network datasets, whether to improve node classification or to discover nodes whose connectivity does not match the network's overall structure.

5. DISCUSSION

Although our case studies indicate that our system is useful, it has some limitations. One problem is the lack of visual feedback for recommendations, i.e., an explanation of why nodes are recommended. For instance, the system currently does not show how much influence past users have on current recommendations. Thus, we plan to improve the suggestion-aware layout, such that the displayed connectivity will help explain why particular nodes are recommended. Another remark is even though the system allows for any N number of recommendations, we find that an ideal number is between 5 and 10. If N is set too low, the user is presented with too few choices, while larger values of N quickly become overwhelming and cause visual clutter. Also, routing outer edges around the top right and lower left of the selection history may not be the best use of space. We plan to try routing all outer edges around the lower left of the navigation path, which would allow us to use the upper right of the screen to display branching selection histories. Another problem is the inefficiency of computing eigenvector centrality. Currently, we treat the centrality computation as a preprocessing step prior to visualization. We plan to create a GPU-based implementation to accelerate computation, which may make it possible to calculate centrality in real time. Finally, since we believe that our system is useful and effective for network exploration, we plan to make it available for general usage.

6. CONCLUSION

We have developed an interactive network visualization system that suggests directions for further navigation based on past user interaction and calculated node importance. Our design greatly simplifies user interaction by only displaying nodes most relevant to the current selection. In addition, we have devised a suggestion-aware layout which optimally positions nodes and recommendations during navigation. We have demonstrated the usefulness and effectiveness of this system by conducting three case studies. First, we used the Genealogy of Influence dataset to show the usefulness of drawing implicit relations in our visualization. Second, we used the VAST 2008 Challenge dataset to show how a user can easily and intuitively leverage our system to find a hidden structure in a large network. Finally, we used the Political Blogosphere dataset to show how our system can find abnormalities in network data that are not apparent from an initial overview.

Acknowledgments

This research was supported in part by the HP Labs, AT&T Labs Research, and U.S. National Science Foundation through grants CCF-0808896, CNS0716691, CCF 0811422, CCF 0938114, and CCF1025269.

References

[AG05] ADAMIC L. A., GLANCE N.: *The political blogosphere and the 2004 U.S. election.* ACM Press, New York, New York, USA, 2005. 8, 9

[BM03] BATAGELJ V., MRVAR A.: Pajek - analysis and visualization of large networks, 2003. 2

[BP98] BRIN S., PAGE L.: The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*, 1-7 (Apr. 1998), 107–117. 2, 3

[Cas91] CASNER S. M.: Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics 10*, 2 (1991), 111–151. 1

[CCM10] CORREA C., CRNOVRSANIN T., MA K.-L.: Visual Reasoning about Social Networks using Centrality Sensitivities. *IEEE transactions on visualization and computer graphics* (Dec. 2010). 3

[CR09] CARENINI G., RIZOLI L.: A multimedia interface for facilitating comparisons of opinions. In *international conference on Intelligent user interfaces* (2009), pp. 325–334. 2

[CZQ*07] CUI W., ZHOU H., QU H., WONG P. C., LI X.: Geometry-Based Edge Clustering for Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2007), 1277–1284. 2

[DCW11] DÖRK M., CARPENDALE S., WILLIAMSON. C.: EdgeMaps: Visualizing Explicit and Implicit Relations. In *Proceeding of Conference on Visualization and Data Analysis* (2011). 4

[EDGH08] ELMQVIST N., DO T.-N., GOODELL H., HENRY N.: ZAME: Interactive Large-Scale Graph Visualization. In *IEEE Pacific Visualization Symposium* (2008), pp. 215–222. 2

[EF09] ELMQVIST N., FEKETE J.-D.: Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics 16*, 3 (2009), 439–454. 2

[FER91] FRUCHTERMAN T. M. J., EDWARD, REINGOLD E. M.: Graph Drawing by Force-directed Placement, 1991. 4, 5

[Fur86] FURNAS G. W.: Generalized fisheye views. In *Human Factors in Computing Systems CHI* (1986), pp. 16–23. 1, 2

[GOB*10] GRETARSSON B., O'DONOVAN J., BOSTANDJIEV S., HALL C., HÖLLERE T.: SmallWorlds: Visualizing Social Recommendations. *Computer Graphics Forum 29*, 3 (2010), 833–842. 1

[GW09] GOTZ D., WEN Z.: Behavior-Driven Visualization Recommendation. In *international conference on Intelligent user interfaces* (2009), pp. 315–324. 1

[HB05] HEER J., BOYD D.: Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization* (2005), pp. 32–39. 2

[HJ07] HACHUL S., JÜNGER M.: Large-graph layout algorithms at work: An experimental study. *Journal of Graph Algorithms and Applications 11*, 2 (2007), 345–369. 2

[IEE] IEEE: Visual Analytics Science and Technology 2008 Grand Challenge. \url{http://www.cs.umd.edu/hcil/VASTchallenge08/}. 6

[Kei02] KEIM D. A.: Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphic 8*, 1 (2002), 1–8. 2

[Kle99] KLEINBERG J. M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM 46*, 5 (Sept. 1999), 604–632. 2, 3

[KSC*08] KOOP D., SCHEIDEGGER C. E., CALLAHAN S. P., FREIRE J., SILVA C. T.: VisComplete: Automating Suggestions for Visualization Pipelines. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1691–1698. 1, 2

[Lov] LOVE M.: Genealogy of Influence. \url{http://mike-love.net/genealogy/}. 6

[LSY03] LINDEN G., SMITH B., YORK J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing 7* (2003), 76–80. 1, 3

[Mac86] MACKINLAY J.: Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics 5*, 2 (1986), 110–141. 1

[MCH*09] MOSCOVICH T., CHEVALIER F., HENRY N., PIETRIGA E., FEKETE J.-D.: Topology-aware navigation in Large Networks. In *International conference on Human factors in computing systems* (2009), pp. 2319–2328. 2

[New07] The mathematics of networks. 1–12. 5

[PS08] PERER A., SHNEIDERMAN B.: Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In *Proceedings of the international conference on Intelligent user interfaces* (2008). 2

[RKMG94] ROTH S. F., KOLOJEJCHICK J., MATTIS J., GOLDSTEIN J.: Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1994), pp. 112–117. 1

[Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics 11*, 1 (1992), 92–99. 2

[SMER06] SHEN Z., MA K.-L., ELIASSI-RAD T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction, 2006. 2

[TLM05] TZENG F.-Y., LUM E. B., MA K.-L.: An Intelligent System Approach to Higher-Dimensional Classification of Volume Data. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 273–284. 1, 2

[TM05] TZENG F.-Y., MA K.-L.: Intelligent Feature Extraction and Tracking for Visualizing Large-Scale 4D Flow Simulations. In *ACM/IEEE conference on Supercomputing* (2005), p. 6. 2

[vHP09] VAN HAM F., PERER A.: Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 953–960. 2

[vHvW05] VAN HAM F., VAN WIJK J. J.: Interactive Visualization of Small World Graphs. In *IEEE Symposium on Information Visualization* (2005), pp. 199–206. 2

[vLKS*10] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual Analysis of Large Graphs. In *Proceedings of Euro- Graphics: State of the Art Report (2010)* (2010). 2

[WZ08] WEN Z., ZHOU M. X.: An optimization-based approach to dynamic data transformation for smart visualization. In *International Conference on Intelligent User Interfaces* (2008), pp. 70–79. 1

[ZC03] ZHOU M. X., CHEN M.: Automated generation of graphic sketches by example. In *Proceedings of the international joint conference on Artificial intelligence* (2003), pp. 65–71. 1

[ZHP*06] ZHOU M. X., HOUCK K., PAN S., SHAW J., AGGARWAL V., WEN Z.: Enabling context-sensitive information seeking. In *International Conference on Intelligent User Interfaces* (2006), pp. 116–123. 1