Social Networks 37 (2014) 56–64

Contents lists available at ScienceDirect

Social Networks

journal homepage: www.elsevier.com/locate/socnet

Visualization techniques for categorical analysis of social networks with multiple edge sets

Tarik Crnovrsanin[a,*], Chris W. Muelder[a], Robert Faris[b], Diane Felmlee[c], Kwan-Liu Ma[a]

[a] Department of Computer Science, 2063 Kemper Hall, University of California-Davis, One Shields Avenue, Davis, CA 95616, United States
[b] Department of Sociology, 2247 Social Science and Humanities, University of California-Davis, One Shields Avenue, Davis, CA 95616, United States
[c] Department of Sociology, Pennsylvania State University, Oswald Tower, University Park, PA 16802, United States

ARTICLE INFO

Keywords:
Social networks
Information visualization
Network analytics
Graph layouts

ABSTRACT

The growing popularity and diversity of social network applications present new opportunities as well as new challenges. The resulting social networks have high value to business intelligence, sociological studies, organizational studies, epidemical studies, etc. The ability to explore and extract information of interest from the networks is thus crucial. However, these networks are often large and composed of multi-categorical nodes and edges, making it difficult to visualize and reason with conventional methods. In this paper, we show how to combine statistical methods with visualization to address these challenges, and how to arrange layouts differently to better bring out different aspects of the networks. We applied our methods to several social networks to demonstrate their effectiveness in characterizing the networks and clarifying the structures of interest, leading to new findings.

Published by Elsevier B.V.

1. Introduction

Social network research is one of the fastest growing academic areas (Rivera et al., 2012) and it continues to expand within an array of social, physical, and biological sciences. One key element of this field of research is social network visualization, which refers to the use of "sociograms," or illustrative diagrams of the ties that connect actors in social networks. The use of graphical representations is one of the main defining properties of the field of social networks (Linton, 2004). While statistical metrics can more succinct, the right metric must be applied. It can be difficult to know a priori what metric will produce the right result, and it can be difficult to verify that the results are correct. Researchers use pictorial images of social networks to help successfully communicate and understand the content of the network and also to aid in uncovering novel, structural patterns within social networks, as well as to guide and confirm statistical metrics. Nevertheless, visual diagrams of social networks often suffer from a range of problems, the most common of which being the high density of edges and complex structures in large networks, yielding sociograms that often appear as indecipherable clouds of nodes and edges.

In the study of aggression networks (Faris and Felmlee, 2011), we identified visualization techniques that can address problems typical to social network visualization, and enhanced the techniques to improve clarity and highlight key structural elements of aggression network. In particular, we considered social networks composed of nodes that can be grouped categorically (i.e., students can be categorized by gender, grade, etc.). Similarly, the edges in a social network can often be divided according to categories (e.g. a friendship is different from an aggression relationship). We used the most common type of visualization, which directly represents relationships between actors as a node-link diagram. That is, the resulting sociograms represent actors with the use of points, or vertices, and the relationships between these actors with the use of lines, or edges, that connect these points. In this paper, we present several visualization techniques tailored to further analyze such social networks. We show how we incorporate statistical measures such as sensitivity analysis to filter nodes/edges from a node-link diagram leading to succinct visualizations, and how different layout designs help bring out structures of interest that would otherwise hidden. We demonstrate several enhanced visualization techniques that enable us to better understand and explain our empirical social network data, and also derive new findings.

2. Related work

Visual diagrams of network-related data have a lengthy history (Linton, 2004). The origins of their application in social network research began with the work of Moreno and Jennings in the early 1930s, in which the typical focus was on examining patterns of individuals' likes and dislikes (Moreno, 1953). Since those early

* Corresponding author. Tel.: +1 916 995 2025.
E-mail addresses: tecrnovr@ucdavis.edu, turokhunter@gmail.com (T. Crnovrsanin).

0378-8733/$ – see front matter. Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.socnet.2013.12.002

beginnings, the use of network visualization has developed into its own specialty field (Freeman, 2000; Brandes and Pich, 2006). This area of study currently receives substantial attention in a range of disciplines (Freeman et al., 1998; Brandes et al., 2001; Di Battista et al., 1994; Krempel et al., 2003; McGrath and Blythe, 2004). Considerable recent work focuses on dynamic visualizations of change in social networks over time (e.g. Moody et al., 2005; Demoll and Mcfarland, 2005; Brandes, 2011; Sallaberry et al., 2012). Other studies examine methods of producing images of web-based social networks (Heer and Boyd, 2005). Finally, researchers continue to develop novel software programs for the production of network images, each with its own unique characteristics, and several publications illustrate the implementation of novel software routines (Brandes, 2011; Heer and Boyd, 2005).

One key task in creating visual images of networks is to determine the appropriate geometrical layout of the nodes and edges. There are several well-defined criteria for assessing the accuracy and validity of a particular graph layout (Demoll and Mcfarland, 2005). Some common criteria (Brandes, 2011; Bertin, 1983) include:

1 edges of the same approximate length,
2 vertices distributed over the area, or
3 reduction of the number of edge crossings.

Nevertheless, optimization of such criteria can be intractable and often contradictory (Brandes, 2011). For surveys of many modern graph layout algorithms see Tollis et al. (1999) or Hachul and Jünger (2005).

The most traditional and commonly used layout algorithm for social network analysis are force-directed layouts (Kamada and Kawai, 1989), often referred to as "spring embedders" (Eades, 1984). In this well-known procedure, nodes in a network graph are positioned iteratively, where the edges connecting them are treated like springs that push and pull on them until the system converges to an equilibrium. By directly optimizing on these criteria, force-directed layouts aim both to distribute nodes widely in a two-dimensional space, and to simultaneously keep connected nodes relatively close to each other.

However, spring embedder techniques do not always scale nicely to large graphs (Brandes and Pich, 2009). Other approaches have been developed with the goal to improve network layout in terms of quality and algorithmic efficiency, especially for large graphs. One such technique (Brandes, 2011) is based on a variant of dimension-reduction methods, referred to as multidimensional scaling (Cox and Cox, 2001), in which the goal is to minimize stress. In this approach, the purpose of stress minimization is to determine positions for every node such that the Euclidean distances in the n-dimensional space resemble the given "dissimilarities" between the nodes, where dissimilarity is determined by graph-theoretic distances, such as the shortest paths (i.e. geodesics).

A fundamental problem that faces visualization of very large social networks, particularly those that use force-directed layouts, is that they often result in a tangled mess of incomprehensible lines; this is often referred to as the "hair-ball" problem. In this paper, we describe two analytic approaches to reduce clutter and produce cleaner network visualizations. First, in order to simplify the contents of a social network, we employ a type of sensitivity analysis that is based on commonly used, graph theoretic, network centrality measures. The findings from the sensitivity analysis (Correa et al., 2012) are then used in traditional graph layouts and node-link diagrams. Second, we employ a type of hierarchical clustering procedure called modularity clustering (Clauset et al., 2004) in order to create an abstraction of a network that is particularly useful in identifying higher level structures.

In this paper, we also show how to apply these analytic strategies in the application of three visual design techniques. The first technique is referred to as "edge bundling" (Danny, 2006). This technique routes similar edges together, which produces cleaner network displays. Next, we introduce a radial layout design that can effectively separate a graph into sub-groups, or communities, for an effective display of network sub-structure. Finally, we introduce the use of "n-partite network layouts" based on parallel coordinate diagrams, which we use to directly compare two or more distinct graphs or subgraphs, defined on the same set of nodes.

In the subsequent two sections, we introduce the techniques we chose to use and explain why and how we enhance them for achieve our goals. Then in the following section, we focus on the study of an aggression network dataset using these techniques. Here we investigate patterns of aggression and friendship among high school students and use visual sociograms to help address questions such as the following:

1 Which students are most likely to be the aggressors, and victims of aggression – those located on the periphery of the friendship network, or those located more centrally?
2 Are there differences by structural factors in patterns of aggression, such as grade level, gender, and race?
3 Do the bulk of aggressive ties occur among or between racial groups?

The techniques that we use are designed to visually reveal the answers to these types of questions.

3. Analysis techniques

To reduce clutter and produce cleaner network visualizations, we apply two analytic approaches. First, we show the use of centrality sensitivity analysis, which measures the importance of one node with respect to another. The aim of this technique is to simplify a network based on centrality metrics, which can then be represented using traditional graph layouts and node-link diagrams. Second, we utilize modularity based clustering, which separates nodes into groups based on the intra and inter group connections. This creates a hierarchical abstraction of a network that we can use to depict higher level structures more clearly.

3.1. Sensitivity analysis

There are four commonly used centrality metrics: Eigenvector (Brin and Page, 1998; Kleinberg, 1999), Markov (White and Smyth, 2003), Betweenness (Lister, 2008; Freeman, 1979), and Closeness (Jacob et al., 2005; Newman, 2003). Each of these measure vertices' overall importance with respect to the whole network. Sensitivity analysis measures a vertex's importance to the structure of the network relative to other vertices in the graph (Correa et al., 2012). This metric is essentially the derivative of centrality, and as such can be calculated similarly for any type of centrality. In this work, we used Eigenvector sensitivity. Eigenvector centrality is a measure of the importance of a node in a network, and is used by the PageRank (Brin and Page, 1998) and Hyperlink-Induced Topic Search (Kleinberg, 1999) algorithms. Rather than basing the importance of a node solely on how many connections it has, eigenvector centrality also takes into account the weights of connections to other nodes; a single connection to a highly important node can carry 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 – in other words, how sensitive the reference node's centrality is to the target node. For instance, if removing a target node's edges results 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 importance relative to the reference node. This can be summarized in the following equation (Correa et al., 2012):

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

where x is eigenvector centrality, t_i is the degree of vertex i, Q is the subtraction of the identity matrix from the adjacency matrix of the network ($Q = AI$), and Q^{+} is the pseudoinverse of Q.

Since every node has a centrality derivative with respect to every other node, centrality sensitivity can be thought of as a complete, weighted network. From this network, we can derive a skeleton network based on edge existence, high centrality derivatives, and overall connectivity (e.g. using a spanning tree). This skeleton network can then be thresholded to be as sparse or as dense as needed, and can be used for a wide variety of purposes, such as simplifying/clarifying layouts, visualizing only the most important connections, or finding important relationships between nodes with no direct connections.

3.2. Modularity clustering

Another way to simplify large, complex networks is to cluster tightly connected groups of nodes together and consider the resulting abstracted super-network. However, trivial application of this approach would completely remove the finer details of the network. Therefore, instead of using a single level of clustering, we employ hierarchical clustering. With hierarchical clustering, the level of clustering can be adjusted dynamically, or multiple clustering levels can be shown at the same time. We employ the "Fast Modularity" clustering algorithm of Clauset et al. (2004), as it is a hierarchical clustering algorithm that has been shown to be effective on real-world networks, and comparable to force directed energy functions (Noack, 2008). Modularity is a metric that evaluates a specific proposed clustering of a network by measuring the density of cluster interiors and the sparsity of inter-cluster connections. Specifically, given a network with a proposed clustering, the modularity Q is defined as:

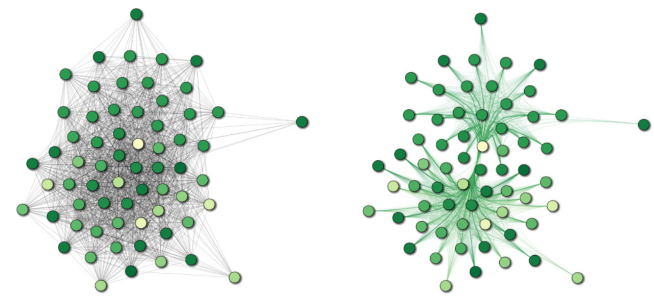
$$Q = \frac{1}{2|E|}\sum_{i,j}\left[A_{i,j} - \frac{k_i k_j}{2|E|}\right]\delta_{i,j}$$

where k_i, k_j are the degrees of nodes i and j, $A_{i,j}$ is 1 if there is an edge between nodes i and j and 0 otherwise, and $\delta_{i,j}$ is 1 if nodes i and j are in the same cluster and 0 otherwise.

The "Fast Modularity" clustering algorithm starts with each node in its own cluster, then greedily merges pairs of clusters such that the change in modularity ΔQ is maximized. In this way, the most tightly connected nodes are clustered together earlier. The end result is a binary hierarchy of clusters, which can be utilized in visualization techniques such as edge bundling (described in Section 4.1).

4. Visualization techniques

Direct visualization of large, complex networks using force-directed layouts often leads to the well-known "hairball problem."



(a) Graph without bundling (b) Graph with bundling

Fig. 1. A graph of the MIT dataset, where each node represents a person and an edge is a call made between individuals. The right images shows how bundling can tease out hidden structure in the graph.

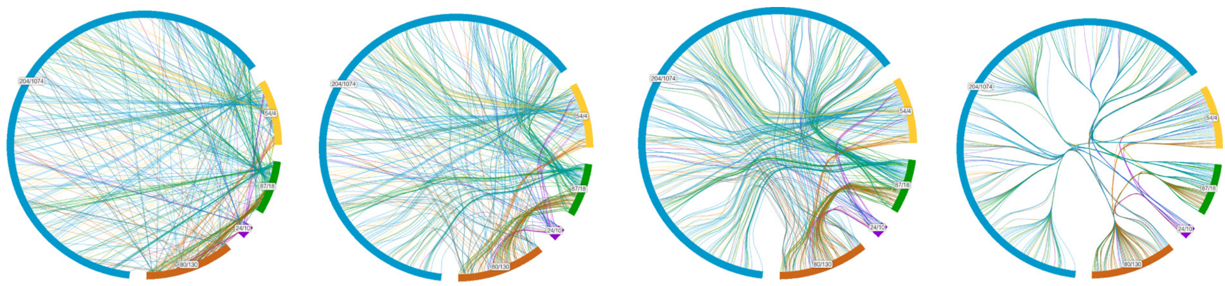
That is, the resulting visualization consists of a tangled mess of incomprehensible lines. We address this problem by introducing enhancements to common visualization techniques. First, we combine the use of modularity clustering and edge bundling to group edges to tease out high-level patterns. Second, we use centrality sensitivity to remove less important edges, at least for the layout process. We also demonstrate a radial layout based design that can effectively separate a graph into sub-groups for comparison, and an n-partite layout that allows comparison between multiple networks.

4.1. Edge bundling

Hierarchical edge bundling (Danny, 2006) is now a well known approach to create cleaner network visualizations that convey high-level patterns without completely sacrificing low-level details. This approach routes edges according to the clustering hierarchy, using cluster centroids as control points for spline curves, as shown in Fig. 1(b). The control points that define an edge between two nodes comprise a path in the clustering hierarchy tree from the first to the second node, passing through their least common ancestor. In this work, we apply this approach using the modularity clustering to construct the tree.

There are a number of different spline models for interpolating between control points. We chose Catmull-Rom and B-spline because Catmull-Rom is computationally simple and offers a high level of control, but B-spline provides smoother lines. This is attributed to Catmull-Rom guaranteeing that the curve goes through the control points while B-spline does not. Another difference is that Catmull-Rom works naturally with as few as two control points, whereas some B-splines require more. For the B-spline, we usually use a B-spline of degree 3. When the number of the control points is 3 or 2, the degree is automatically reduced to 2 or 1, respectively, since it is required that the degree is lower than the number of control points.

We can loosen these tight bundles by introducing a bundling strength. As shown in Fig. 2, the bundling strength is a user-specified parameter $B \in [0, 1]$, and controls the amount of bundling by linearly interpolating between a straight line at $B = 0$ and the spline defined by the path from node to node through the least common ancestor at $B = 1$. This is accomplished by defining a new set of points, evenly spaced along the straight line between the two nodes, then linearly interpolating each control point between the linear point and the corresponding cluster-defined point, and finally rendering the spline according to the interpolated points.



(a) $B = 0.0$ (b) $B = 0.2$ (c) $B = 0.5$ (d) $B = 1.0$

Fig. 2. As the bundle strength increases, the lines interpolate from straight position to the control points. The ideal *B* is usually somewhere between 0.8 and 1 where the lines can be followed and discerned through the bundles.

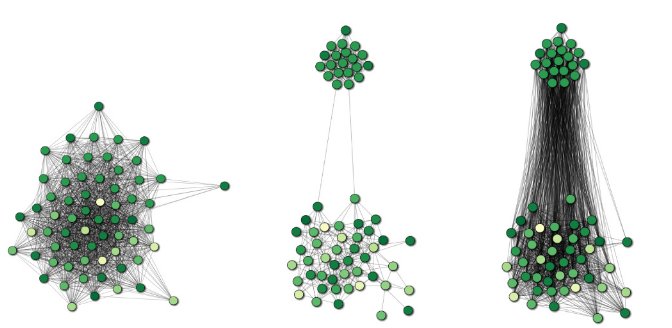
4.2. Sensitivity based layout

In many networks, there are some nodes and edges that are particularly important to the structure of the network, but many that are less so. When a layout algorithm uses all of the edges, the resulting visualization can look like a bird's nest, without any apparent structure beyond simple core-periphery pattern, as shown in Fig. 3(a). By filtering edges based on sensitivity analysis, we can consider a simplified skeleton network that retains the structural properties of the original network. Since this skeleton network is much sparser than the original network, it can be effectively laid out and visualized using a traditional node-link diagram, as shown in Fig. 3(b). The layout of this skeleton network is often much better than a layout defined by the whole network, so we can use the skeleton network's layout and reintroduce the original edges to produce an improved node-link diagram of the entire graph, as shown in Fig. 3(c). This new skeleton network can also be used to improve both modularity clustering by adding weights to the calculation, and edge bundling by routing edges through the more central paths.

4.3. Radial representation

Sometimes, merely improving the layout algorithm is insufficient for showing particular aspects of a network. Specifically, social networks can be divided into groups according to discrete properties besides connectivity, such as gender, race, school grade, or others. However, the density of ties in most traditional node-link diagrams make it difficult to distinguish in inter-group patterns from intra-group patterns. Therefore, we introduce enhancements to a radial representation that arrange nodes according to additional properties as well as connectivity, as shown in Fig. 4. Nodes are placed around a circle, grouped into discrete arcs based on the selected data attribute, and ordered within each group by connectivity with the use of modularity clustering. This new representation also delegates the two kinds of connections to separate regions of space: intra-group edges are displayed outside the circle while inter-group edges are drawn in the middle. The label on each group shows the number of inter-group and intra-group connections, respectively.

To compute a useful ordering for the nodes, we first construct a clustering hierarchy using the modularity of the entire network, as shown in Fig. 5. This tree does not take into account the data attribute of interest, and thus could be pre-calculated. However, nodes that are next to each other in the tree might be in distinct groups based on the property of interest. Therefore, we replicate the clustering tree, creating one for each group, and proceed to filter each tree so that it only contains one group. This is done by traversing the tree and removing any cluster that does not contain at least one leaf node from the desired group. This creates trees that have many nodes with only one child. To reduce the depth of the tree, we trim the tree by also removing intermediate clusters



(a) Normal Graph. (b) Sensitivity Cutoff. (c) Sensitivity Cut-off with all edges.

Fig. 3. (a) Full MIT dataset. (b) Shows the reduced network from our sensitivity cutoff and (c) with all the edges reintroduced.

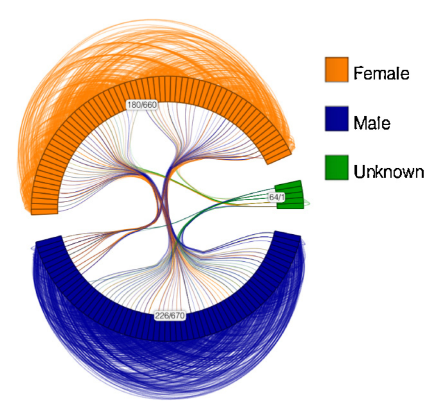


Fig. 4. A simple example of our radial layout approach. Intra-group edges are drawn outside the circle while inter-group edges are drawn through the middle. The color of the edge represents the source of the edge. The first label on each group shows the number of inter-group, while the second number shows the number intra-group. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

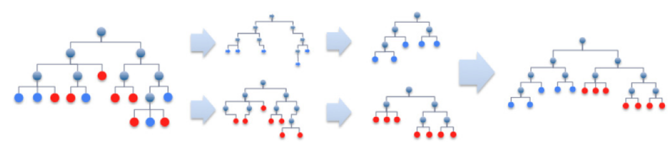


Fig. 5. First, we construct a tree from the network using modularity clustering, where the color of the nodes represents which group they belong to. To do so, we split up the two colors into smaller subtrees. Then the trees are trimmed and combined under a central root node.

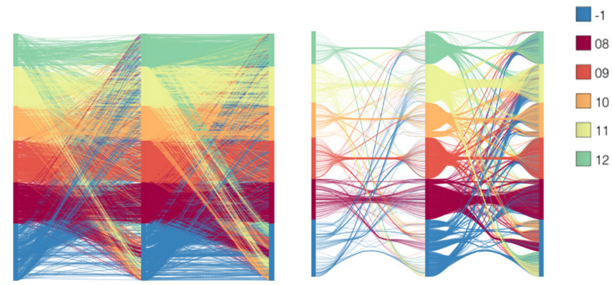
which only have one child (thereby reducing the number of control points needed for the spline code). Once all the trees are trimmed, we link them all together with one root node to create a single tree where each branch from the root represents a sub-tree for an individual group. It would alternatively be possible to run modularity clustering on each group separately to improve the arrangement within groups, but this would ignore inter-group edges and hence hinder the ability to analyze inter-group relationships. Once this clustering is finished, we traverse the tree to generate an ordering which can then be used to arrange nodes around the circle.

In order to visually separate inter-group and intra-group edges, they are drawn in separate regions of space. To connect edges, we modify the edge bundling by defining custom control points for the two types of edges separately. We utilize both previously stated types of splines: B-splines for the exterior edges and Catmull-Rom for interior edges. Inter-group edges are routed through the inside of the circle, so control point locations are calculated in polar coordinates, using weighted centroids for the angles, and the clustering depth for the radius. However, if the calculated route included every node of the tree up through the LCA (least common ancestor), then every edge would go through the center of the circle, since all edges routed through the center of the circle are inter-cluster and thus have to cross the root of the tree. To avoid this issue, we implement a cutoff value, which allows the user to remove control points, providing more distinct bundles between groups. This cutoff can be counted either from the start/end nodes (as number of points to keep) or from the root (as number of nodes to prune). We found that defining the cutoff from the root usually produces better results. For intra-group nodes, we also use polar coordinates, with the angle defined by the weighted centroids and the radius defined by level, but we extend the edges outside the circle instead of the inside. In this case, using the LCA would be acceptable, since each tree is distinct. However, due to the strong curvature of most edges outside the circle, we found it most effective to use only the leaf nodes and the LCA to define control points, bypassing intermediate control points.

4.4. Parallel coordinates

Sometimes networks contain more than one kind of edge, defining 2 or more unique networks on the same set of nodes. In such cases, a layout that is good for one set of edges might not be good for another. Alternately, with one unified layout, sparser networks may get lost inside denser ones. Here, we describe a representation based on n-partite network layouts, where groups of nodes are laid out parallel to each other. We extend this concept to multiple networks on the same set of nodes by replicating the nodes, and considering each network a bipartite graph from the full set of nodes to a duplicate set of nodes, which creates an n-partite network where n is one more than the number of edge sets. We can then lay out this n-partite network in a series of columns by evenly spacing the nodes in each column. Edge directionality is also shown in this representation, since all edges proceed from left to right. While hierarchical layouts such as Sugiyama et al. (1981) or Dwyer and Koren, 2005 could be used instead, we have the unique situation of each group of nodes being identical, and thus it is more natural for each column to have the same ordering. Thus, we reuse the afore calculated modularity clustering to cluster the nodes, and traverse this hierarchy to define a universal ordering.

As shown in an application of this technique in Fig. 6(a), the number of edges between each column can become quite dense. Therefore, we apply edge bundling to clarify the result, as shown in Fig. 6(b). The same modularity clustering that defines the ordering can be utilized, but as in the radial layout, control points need to be computed. Here, the y values of the control points are defined by weighted centroids, but the x coordinates are defined by level. As



(a) Parallel with no bundling (b) Parallel with bundling

Fig. 6. Both parallel coordinates diagrams show aggression network on the left and friendship network on the right. All three axes are ordered the same and are grouped by grade level. By bundling the edges, the diagram on the right is easier to read. It suffer from less visual clutter than the unbundled version. By looking at the diagram on the right we can see that the two networks look very similar, indicating that aggression mimics friendship.

in the radial view, a level cutoff is employed, otherwise all edges would go through the center of each region.

5. A study of a friendship/aggression network datasets

To demonstrate the effectiveness of these techniques, we have applied them to the visualization of two friendship/aggression network datasets.

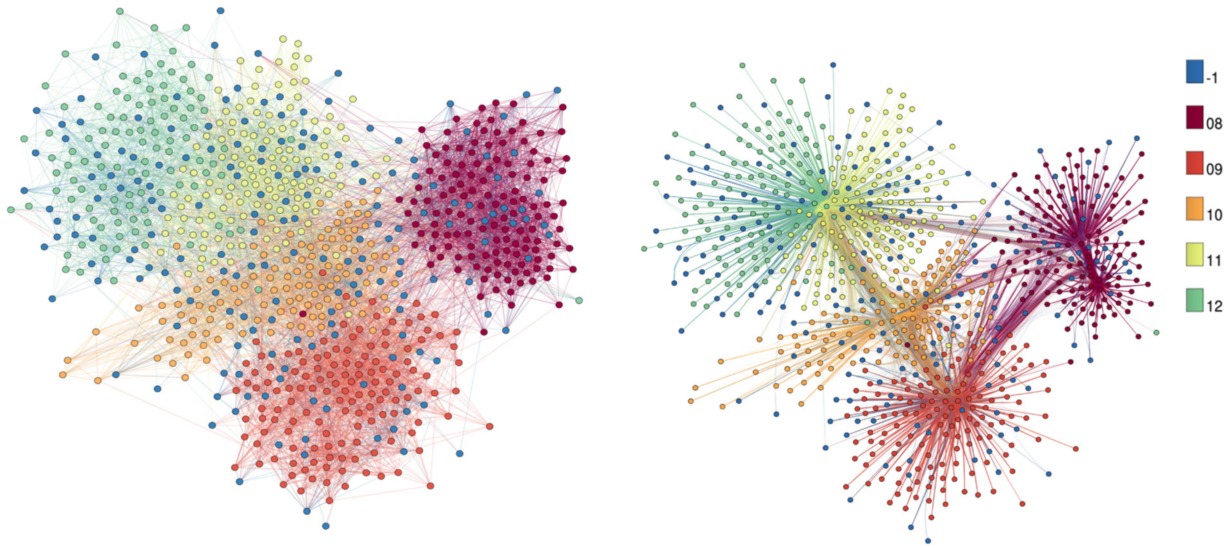
5.1. Data sets

The first dataset, the Contexts of Adolescent Substance Use (hereafter, Contexts), is a longitudinal study of adolescents in three counties in North Carolina. At the start, all students in the 6th, 7th, and 8th grades in all public schools in the three counties were asked to participate. Data on a wide range of health factors and risk behaviors (for a recent example of research using the data, see Ennett et al., 2010) were collected every six months for six waves, followed by a final wave of data collection one year later. At each wave, students were asked to nominate up to five of their closest friends, and after wave 3, were also asked to nominate up to five schoolmates they “picked on or were mean to” and up to five whom they “picked on or were mean to.” Students were instructed to only consider serious incidents of cruelty, and disregard playful teasing. The aggression and victimization networks were combined, such that we considered there to be an aggressive tie from A to B if either A nominated B as a victim, or B nominated A as an aggressor (for research on the aggression network data, Faris and Felmlee, 2011).

The second dataset, the Long Island School Study, was intended to replicate and extend the Contexts data, albeit in a single school and over a shorter time frame. Data were collected on a biweekly basis over a two month period from a single Long Island public school in an affluent suburb of New York City. The nearly exclusively white and highly affluent nature of the Long Island school differentiate it from the disproportionately African-American, rural, and lower socioeconomic status of the Contexts study population. Importantly, students were again asked to nominate friends, victims, and aggressors, using the same language as the Contexts study, but were allowed to nominate more of their peers (10 in the case of friends, and 8 victims and aggressors). See Faris and Felmlee (2012) and Felmlee and Faris (2013) for more information.

5.2. School friendship/aggression networks

When looking at the Long Island friendship dataset using the sensitivity-based force-directed layout and coloring by grade, the 8th grade cluster is distinctly separate from the rest of the grades.



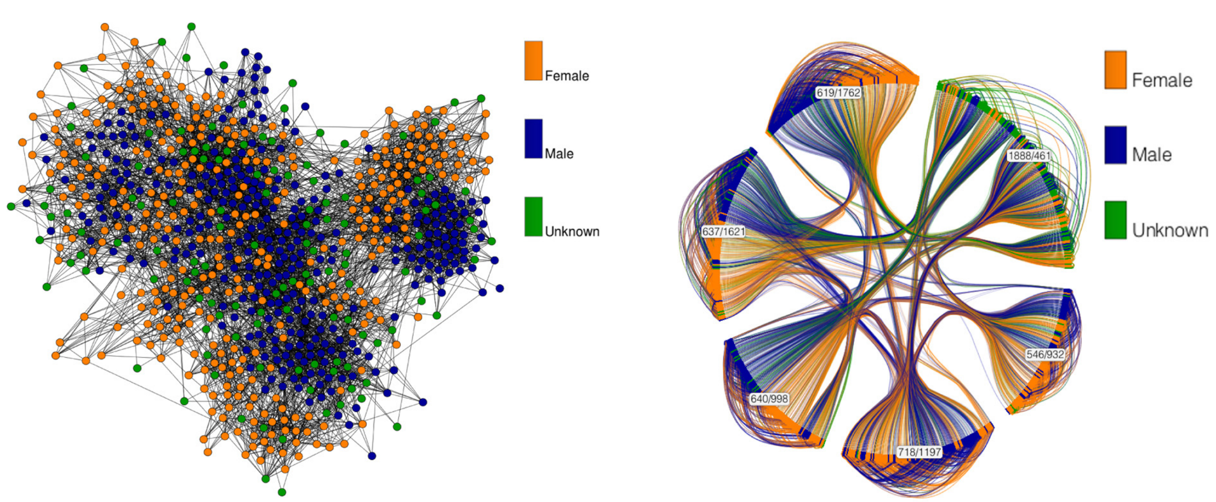
(a) Graph using Eigen Sensitivity cutoff (b) Graph with bundling

Fig. 7. Long Island friendship dataset shown in directed node-link graph. In both graphs, the edges are colored by the source node. Bundling the edges, helps us see the relationship between groups. For instance, 11th and 12th grader are highly connected to each other. Most of the edges from the 8th grader group are red implying that more 8th graders say they are friends with higher grades than vice versa. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

This makes sense, since 8th graders are just entering into the school and thus have not had time to make as many friends outside of their grade. While 9th grade is relatively distinct at the bottom of the graph, 10th, 11th, and 12th graders are harder to distinguish. Applying sensitivity analysis to remove some of the edges from the force calculation results in a cleaner layout, as shown in Fig. 7. All edges are still rendered, maintaining the completeness of the graph. However, there is a better separation between grades, though 11th and 12th grade are still tightly connected to each other. Since the sensitivity cutoff removes less important edges, we know that this pattern is important to the structure graph. This shows that as students get older, age plays less of a role when it comes to friendships.

Fig. 8(a) depicts the Long Island network colored according to gender. In the 8th grade cluster, there is a clear segregation according to gender. In the 9th grade, the males are clustered in the center while females are in small groups around the males. 10th grade also divides somewhat by gender, but 11th and 12th grades show almost no gender segregation.

Our radial layout can show both gender and grade patterns at the same time. In Fig. 8(b), we separate the groups by grade and color the nodes by gender. This allows us to look at the grades completely separately, which we could not do in the node-link graph since 11th and 12th were mixed together. This representation clearly shows that there is more gender mixing as individuals get older. This change happens rather abruptly between the 10th and 11th



(a) Node-link graph colored by gender. (b) Radial layout grouped by grade level and colored by gender.

Fig. 8. The node link graph on the left runs into limitation when trying to compare multiple properties, since only one property can be mapped to color at a time. This makes it hard for the user to look at both gender and grade level. In the radial layout on the right, we group by grade and map color to gender. The visualization starts with 8th grade on top and continues counter-clockwise with 12th grade at bottom right and unknown to the top right. The radial layout shows that gender plays less of a role as kids get older (there is more mixing of gender in higher grades).

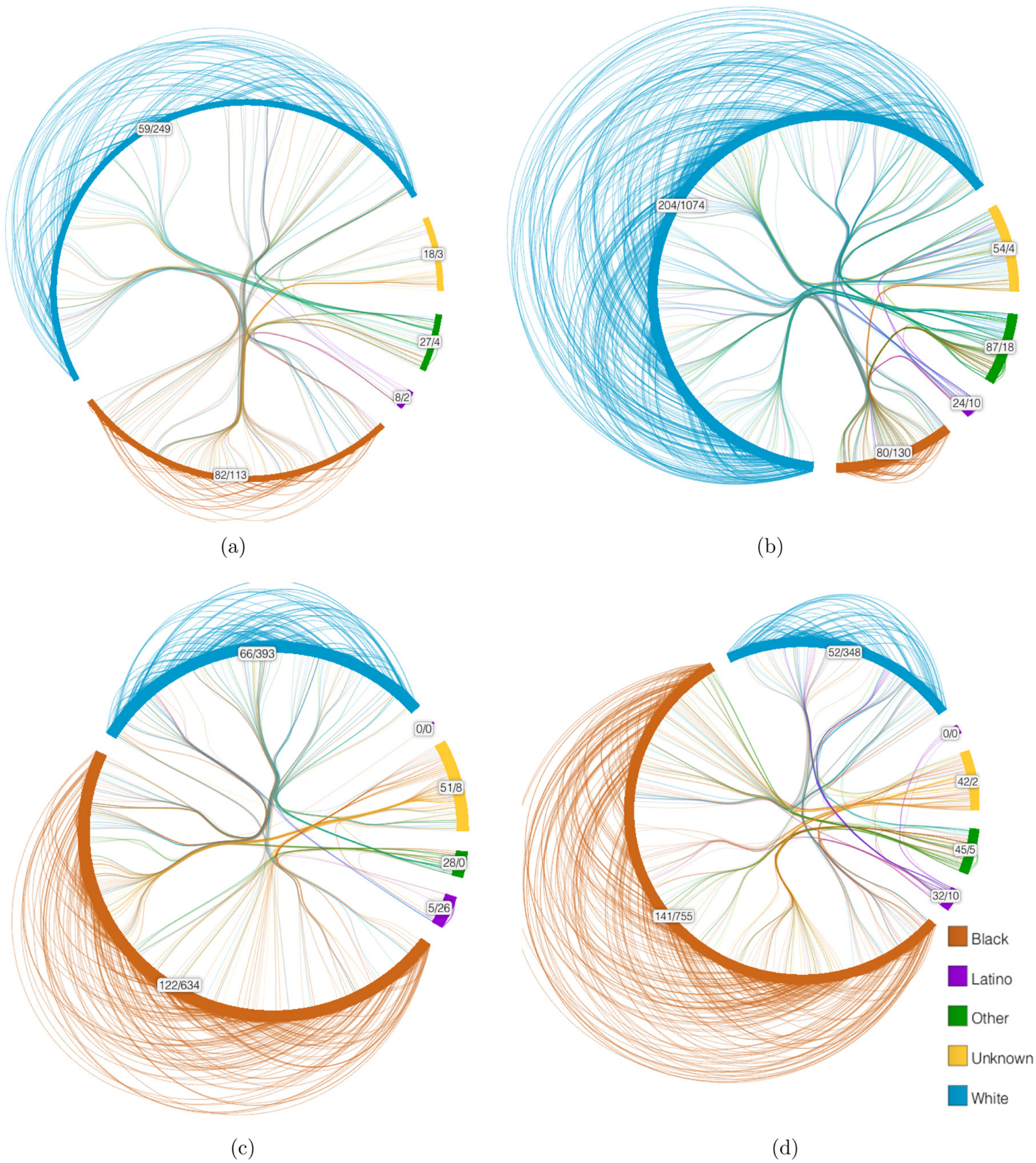


Fig. 9. Four schools from the Contexts dataset showing the aggression network. Each school has a different race breakdown and is grouped and colored by race. The label shows the number of inter- and intra-group connections. Comparing the four graphs, we can see that race does not play a major rule in aggression. A more telling factor is the size of the groups. Bigger groups tend to victimize everyone more or less equally, while minority groups have less internal aggression. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

grades. Also interesting is the fact that modularity clustering has fairly strongly separated the group of unreported grades according to grade levels. Therefore, we predict that the clustering can infer the grades of the students in the unreported group with relatively high accuracy.

We were also interested in finding out if there were any correlations between the friendship and aggression networks. We confirmed that there were. This can be seen using the parallel coordinates view shown in Fig. 6(b), with the friendship network on the right and the aggression network on the left. The ordering for the axis is defined by the modularity clustering of the combined network, albeit separated and colored according to grade level. When the connections are drawn as straight lines, patterns are not invisible due to clutter. But, by applying the edge bundling technique to the connections, the patterns become more clear. In this view, it becomes more readily apparent that the majority of edges stays within grades. Interestingly, the aggression network seems to mimic the friendship network. This at first might seem unintuitive, but if aggression is a way of jockeying for social status (Faris and Felmlee, 2011), then we might hypothesize that much aggression occurs between rivals who may be members of the same social groups.

We wanted to see how race played a role in aggression. Fig. 9 shows several schools from the Contexts dataset, each with a different racial breakdown. We colored and grouped each graph by race. Fig. 9(a) and (c) shows similar acts of aggression with Whites being the majority in one and Blacks in the other. In both cases, members of the predominant race victimize everyone more or less equally,

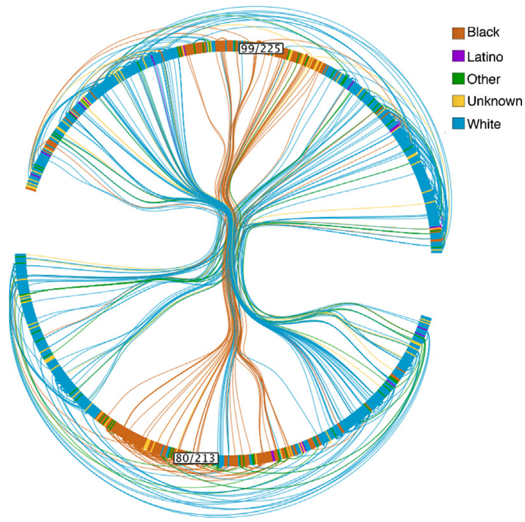


Fig. 10. Aggression radial layout shows one of the Contexts schools grouped by gender and colored by race. The female gender is shown on top. Aggression occurs mostly within the same race, and students generally group with people of the same race.

while the minority group has much less internal aggression. This indicates that race itself does not play a large role in aggression. Rather, aggression behavior is influenced by the relative size of the racial composition. Interestingly, when the school is broken down by gender and colored by race, as seen in Fig. 10, individuals tend to group according to race, and aggression is also apparent primarily internal within races. There also seems to be no difference between gender aggression. Both gender have the same internal and external aggression.

6. Conclusion

We have described how to apply visualization approaches to social network problems, as well as how to enhance them by incorporating statistical measures and tailor different layout methods to particular analysis tasks. In particular, we have demonstrated visualization that can not only utilize standard statistical metrics, but which can be used to select appropriate statistical metrics, evaluate or confirm their results, or in some cases even improve them (e.g. in the case of missing data).

These approaches have been applied to the analysis of networks with multiple categorical breakdowns, both in node categories and edge categories. Social networks of all varieties are patterned by the categorical distinctions among their members, the most well-known being homophily according to demographic characteristics. These visualization techniques readily reveal patterns that are difficult to discern with traditional visualization approaches.

Moreover, we have addressed the “hairball problem” endemic to dense networks that are typically visualized with spring-embedder algorithms. Critically, our sensitivity cutoff approach reveals structural properties that would have otherwise remained invisible. While no single approach can succinctly reveal everything about any one large network, our approaches provide useful tools for social network analysis.

Acknowledgements

We thank James Moody for his helpful comments. This work is sponsored in part by the UC Davis RISE program, HP Labs, and the U.S. National Science Foundation through Grants CCF 1025269 and CCF 1320229.

References

Bertin, J., 1983. Semiology of Graphics. University of Wisconsin Press.
Brandes, U., Pich, C., 2007. Eigensolver methods for progressive multidimensional scaling of large data. In: Proceedings of the 14th International Conference on Graph Drawing, GD'06, pp. 42–53.
Brandes, U., Pich, C., 2009. More flexible radial layout. In: Graph Drawing., pp. 107–118.
Brandes, U., Raab, J., Wagner, D., 2001. Exploratory network visualization: simultaneous display of actor status and connections. Journal of Social Structure 2 (4).
Brandes, U., Indlekofer, N., Mader, M., 2011. Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. Social Networks 34 (3), 291–308.
Brin, S., Page, L., 1998. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30 (April (1–7)), 107–117.
Clauset, A., Newman, M.E.J., Moore, C., 2004. Finding community structure in very large networks. Physical Review E, 1–6.
Correa, C., Crnovrsanin, T., Ma, K.-L., 2012. Visual reasoning about social networks using centrality sensitivity. IEEE Transactions on Visualization and Computer Graphics 18 (January (1)), 106–120, IEEE Educational Activities Department.
Cox, T., Cox, M., 2001. Multidimensional Scaling. Chapman & Hall, London.
Danny, H., 2006. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics 12, 741–748, IEEE Computer Society.
Demoll, B.S., Mcfarland, D., 2005. The art and science of dynamic network visualization. JoSS: Journal of Social Structure 7.
Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G., 1994. Algorithms for drawing graphs: an annotated bibliography. Computational Geometry: Theory and Applications 4 (5), 235–282.
Dwyer, T., Koren, Y., 2005. DIG-COLA: directed graph layout through constrained energy minimization. In: IEEE Symposium on Information Visualization, pp. 65–72.
Ennett, S., Vangie, F., Karl, B., Andrea, H., Robert, F., John, H., Li, C., 2010. A social contextual analysis of youth cigarette smoking development. Nicotine & Tobacco Research 12, 950–962.
Eades, P., 1984. A heuristic for graph drawing. Congressus Numerantium 42, 149–160, Utilitas Mathematica Publishing; Winnipeg, Canada.
Faris, R., Felmlee, D., 2011. Status struggles network centrality and gender segregation in same- and cross-gender aggression. American Sociological Review 76 (1), 48–73.
Faris, R., Felmlee, D., 2012. Social Networks and Aggression at the Wheatley School. http://i2.cdn.turner.com/cnn/2011/images/10/10/findings.from.the.wheatley.school.pdf
Felmlee, D., Faris, R., 2013. Toxic ties: networks of friendship, dating, and cyber victimization. In: Presented at the Annual Meetings of the American Sociological Association.
Freeman, L.C., Webster, C.M., Kirke, D.M., 1998. Exploring social structure using dynamic three-dimensional color images. Social Networks 20, 109–118.
Freeman, L.C., 1979. Centrality in social networks conceptual clarification. Social Networks 1 (3), 215–239, Elsevier.
Freeman, L.C., 2000. Visualizing social networks. Journal of Social Structure 1 (1).
Hachul, S., Jünger, M., 2005. An experimental comparison of fast algorithms for drawing general large graphs. In: Graph Drawing, pp. 235–250.
Heer, J., Boyd, D., 2005. Vizster: visualizing online social networks. In: INFOVIS'05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization, Washington, DC, USA, 2005. IEEE Computer Society, p. 5.
Holten, D., 2006. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics 12 (5), 741–748.
Jacob, R., Koschützki, D., Lehmann, K., Peeters, L., Tenfelde-Podehl, D., 2005. Algorithms for centrality indices. In: Network Analysis, Lecture Notes in Computer Science, vol. 3418. Springer, Berlin Heidelberg, pp. 62–82.
Kamada, T., Kawai, S., 1989. An algorithm for drawing general undirected graphs. Information Processing Letters 31 (April (1)), 7–15.
Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (September (5)), 604–632.
Krempel, L., Plümper, T., 2003 February. Exploring the dynamics of international trade by combining the comparative advantages of multivariate statistics and network vizualizations. Journal of Social Structure 4 (1), 1–22.
Linton, C., 2004. The Development of Social Network Analysis: A Study in the Sociology of Science. Booksurge.
Lister, R., 2008. After the gold rush: toward sustainable scholarship in computing. In: Simon, Hamilton, M. (Eds.), Tenth Australasian Computing Education Conference (ACE 2008), CRPIT, vol. 78. NSW, Australia. ACS, pp. 3–18.
McGrath, C., Blythe, J., 2004. Do you see what I want you to see? the effects of motion and spatial layout on viewers' perceptions of graph structure. Journal of Social Structure 5 (2).
Moody, J., McFarland, D., Bender-deMoll, S., 2005. Dynamic network visualization. American Journal of Sociology 110 (January (4)), 1206–1241.
Moreno, J.L., 1953/1978. Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama.
Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Review 45 (2), 167–256.

Noack, A., 2008. Modularity clustering is force-directed layout. CoRR (abs/0807.4052).
Rivera, M.T., Soderstrom, S.B., Uzzi, B., 2012. Dynamics of Dyads in Social Networks: Assortative, Relational, and Proximity Mechanisms., pp. 91–115.
Sallaberry, A., Muelder, C.W., Ma, K.-L., 2012. Clustering, visualizing, and navigating for large dynamic graphs. In: Proceedings of Proceedings of Graph Drawing 2012.
Sugiyama, K., Tagawa, S., Toda, M., 1981. Methods for visual understanding of hierarchical systems. IEEE Transactions on Systems, Man, and Cybernetics 11, 109–125.
Tollis, I.G., Di Battista, G., Eades, P., Tamassia, R., July 1999. Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall.
White, S., Smyth, P., 2003. Algorithms for estimating relative importance in networks. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'03, pp. 266–275.