

# Spectrum-Based Network Visualization for Topology Analysis

Xianlin Hu, Aidong Lu, and Xintao Wu ■ *University of North Carolina at Charlotte*

**S**ocial networks often embed rich topological information, such as community structures. The topology structure includes the individual nodes' and edges' relationships and their importance to the communities. Visualizing and

---

**Based on recent achievements in spectrum-based analysis, a visual-analytics approach uses the features of node distribution and coordinates in the high-dimensional spectral space. To assist exploration of network topologies, network visualization and interactive analysis let users filter nodes and edges in a way that's meaningful to the global topology structure.**

navigating the network topology are crucial to understanding various networks.

Although directly visualizing network topology is extremely challenging, you can express many graph topological features as an explicit function of a spectrum and eigenvectors. A network's eigenvalues are intimately connected to many important topological features.<sup>1</sup> For example, an adjacency matrix's eigenvalues encode information about a network's cycles and diameter. The maximum degree, the chromatic number, the clique number, and

the extent of branching in a connected graph are all related to the largest eigenvalue. In this article, we study useful spectrum features and explore how to incorporate them to develop topology-aware network visualizations.

We've developed a visualization approach for analyzing network topologies. We mainly study undirected and unweighted networks without self-loops, because they're challenging to analyze and we can decompose other network representations

to this format. In our approach, an automated network layout algorithm uses the features of node distribution and spectral-space coordinates. We also provide interactive tools for analyzing topologies. Case studies and comparisons demonstrate the advantages of our spectrum-based approach, which differs significantly from previous force- or energy-directed approaches. (For a look at related work in network visualization and spectrum-based approaches, see the sidebars.)

## Automated Network Layout

Here we examine our approach's foundation—spectrum-based network analysis—and describe our network layout algorithm's main steps and performance.

## Network Analysis

A network or graph  $G(V, E)$  is a set of  $n$  nodes  $V$  connected by a set of  $m$  links  $E$ , where  $E \subseteq V \times V$ . We can represent  $G$  as a symmetric adjacency matrix  $A = (a_{ij})_{n \times n}$ . Because we concentrate here on analyzing data with binary adjacency matrices,  $a_{ij} = 1$  if node  $i$  is connected to node  $j$ ; otherwise,  $a_{ij} = 0$ .

Graph spectral analysis deals with the nodes' spectra (eigenvalues and eigenvector components). Let  $\lambda_i$  be the eigenvalues of  $A$  and  $x_i$  be the corresponding eigenvectors. When  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ , the spectral decomposition of  $A$  is  $A = \sum_i \lambda_i x_i x_i^T$ . Let  $x_{ij}$  denote the  $j$ th entry of  $x_i$ . We get this adjacency matrix:

## Related Work in Network Visualization

Network visualization is a powerful, widely used tool. Researchers have applied its techniques to various fields, including personal social networks, telecommunication networks, citation networks, computer file systems, and the Web. Popular network visualization approaches include node links, space division, space nested visualization, and matrix visualization.<sup>1</sup> In particular, using node-link graphs to visualize networks is common because it's a natural representation of network topology.

One core problem in node-link network visualization is generation of the network layout, which provides the foundation for network visual analysis and exploration techniques. We can categorize these algorithms according to the types of network layouts or mechanisms. For example, force-directed approaches often simulate a network as a physical system in which edges are springs and nodes are particles.<sup>2</sup> Andreas Noack proposed the LinLog method to separate communities by minimizing an energy model iteratively.<sup>3</sup>

Also, owing to data's increasing size and complexity, visualization researchers have developed interactive analysis approaches. For example, Yu-Hsuan Chan and her colleagues provided a set of interaction methods for analyzing graph hierarchies.<sup>4</sup>

### References

1. W. Cui, "A Survey on Graph Visualization," Hong Kong Univ. of Science and Technology, 2007.
2. D. Auber et al., "Multiscale Visualization of Small World Networks," *Proc. 9th Ann. IEEE Conf. Information Visualization (INFOVIS 03)*, IEEE CS, 2003, pp. 75–81.
3. A. Noack, "An Energy Model for Visual Graph Clustering," *Proc. 11th Int'l Symp. Graph Drawing (GD 03)*, LNCS 2912, Springer, 2003, pp. 425–436.
4. Y.-H. Chan, K. Keeton, and K.-L. Ma, "Interactive Visual Analysis of Hierarchical Enterprise Data," *Proc. IEEE Int'l Conf. E-commerce Technology*, IEEE, 2010, pp. 180–187.

$$\alpha_u \rightarrow \begin{pmatrix} x_1 & & x_i & & x_k & & x_n \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ x_{11} & \cdots & x_{i1} & \cdots & x_{k1} & \cdots & x_{n1} \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_{1u} & \cdots & x_{iu} & \cdots & x_{ku} & \cdots & x_{nu} \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_{1n} & \cdots & x_{in} & \cdots & x_{kn} & \cdots & x_{nn} \end{pmatrix}, \quad (1)$$

where  $x_i$  is a column vector and the row vector  $\alpha_u(x_{1u}, x_{2u}, \dots, x_{nu})$  represents the coordinates of node  $u$  in the  $n$ -dimensional spectral space.

An intimate relationship exists between the combinatorial characteristics of a graph and the algebraic properties of its adjacency matrix. Xiaowei Ying and Xintao Wu explored the relationships of spectral coordinates and communities.<sup>2</sup> They loosely defined communities as collections of network nodes that interact unusually frequently. In particular, they proposed the *quasi-orthogonal property*:

For a graph with  $k$  communities, the coordinate of node  $u$  in  $k$ -dimensional space,  $\alpha_u = (x_{1u}, x_{2u}, \dots, x_{ku}) \in \mathbb{R}^{1 \times k}$ , denotes the likelihood of node  $u$ 's attachment to these  $k$  communities. Node points within one community form a line that goes through the origin in the  $k$ -dimensional space. Nodes in  $k$  communities form  $k$  quasi-orthogonal lines in the spectral space.<sup>2</sup>

To prove this theorem, you optimize graph division, which maximizes the overall edge-to-node

densities in each divided community. The maximum density reaches the sum of all the eigenvalues only when the quasi-orthogonal property is ensured.<sup>2</sup>

To demonstrate the relationship between eigenvectors and network topology, we used a political-book network with 105 nodes and 441 edges (see Figure 1a).<sup>3</sup> Nodes represented books about US politics sold by Amazon.com; edges represented frequent copurchasing of books by the same buyers. Each node was labeled "liberal" (blue), "conservative" (red), or "neutral" (white). Mark Newman assigned the labels after reading the books' descriptions and reviews posted on Amazon.com.<sup>4</sup>

Figure 1b plots node coordinates projected in the 2D spectral space of  $A$ . Most vertices are distributed along two straight, quasi-orthogonal lines. This indicates that two communities exist, connected by sparse edges. The upward-trending line consists mostly of red nodes; the downward-trending line consists mostly of blue nodes. White nodes, which correspond to either noise or bridging nodes, are distributed either around the origin or between two quasi-orthogonal lines in the projected space.

### Network Layout

Our algorithm comprises *node projection*, *node dispersion*, and *sphere warping* (see Figure 2).

**Node projection.** Given a user-assigned parameter  $k$  (indicating the number of dimensions in the spectral space we're going to use), we calculate  $\lambda_1$  and  $x_i$  for  $A$ . As Equation 1 shows, the coordinates

## Related Work in Spectrum-Based Approaches

In applied mathematics and scientific computing, spectrum-based approaches have demonstrated their usefulness as one part of graph theory. Especially for network analysis, researchers have performed spectral-related research to analyze and describe network properties and node relationships. For example, Andrew Seary and William Richards used spectral methods to discover networks' cohesive and localized features.<sup>1</sup> Mark Newman used a matrix's eigenspectrum to detect community structure in networks.<sup>2</sup>

To improve network layouts, you can integrate the data features explored in spectral spaces. For example, Yehuda Koren and his colleagues developed the ACE (Algebraic Multigrid Computation of Eigenvectors) algorithm to draw graphs recursively by eigenvectors of the Laplacian matrix.<sup>3</sup> Pak Chung Wong and his colleagues visualized a power grid network, using an enhanced spectral layout algorithm for accelerating eigenvector computation.<sup>4</sup> They demonstrated this visualization for planning and monitoring tasks.

Unlike previous spectrum-based approaches, ours (see the main article) uses a noniterative algorithm that pro-

duces network layouts by transforming node locations according to their distribution features in high-dimensional spectral spaces.

### References

1. A. Seary and W. Richards, "Spectral Methods for Analyzing and Visualizing Networks: An Introduction," *Nat'l Research Council, Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, Nat'l Academies Press, 2003, pp. 209–228.
2. M.E.J. Newman, "Finding Community Structure in Networks Using the Eigenvectors of Matrices," *Physical Rev. E*, vol. 74, no. 3, 2006, article 036104.
3. Y. Koren, L. Carmel, and D. Harel, "Drawing Huge Graphs by Algebraic Multigrid Optimization," *Multiscale Modeling and Simulation*, vol. 1, no. 4, 2003, pp. 645–673.
4. P.C. Wong et al., "A Novel Visualization Technique for Electric Power Grid Analytics," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 3, 2009, pp. 410–423.

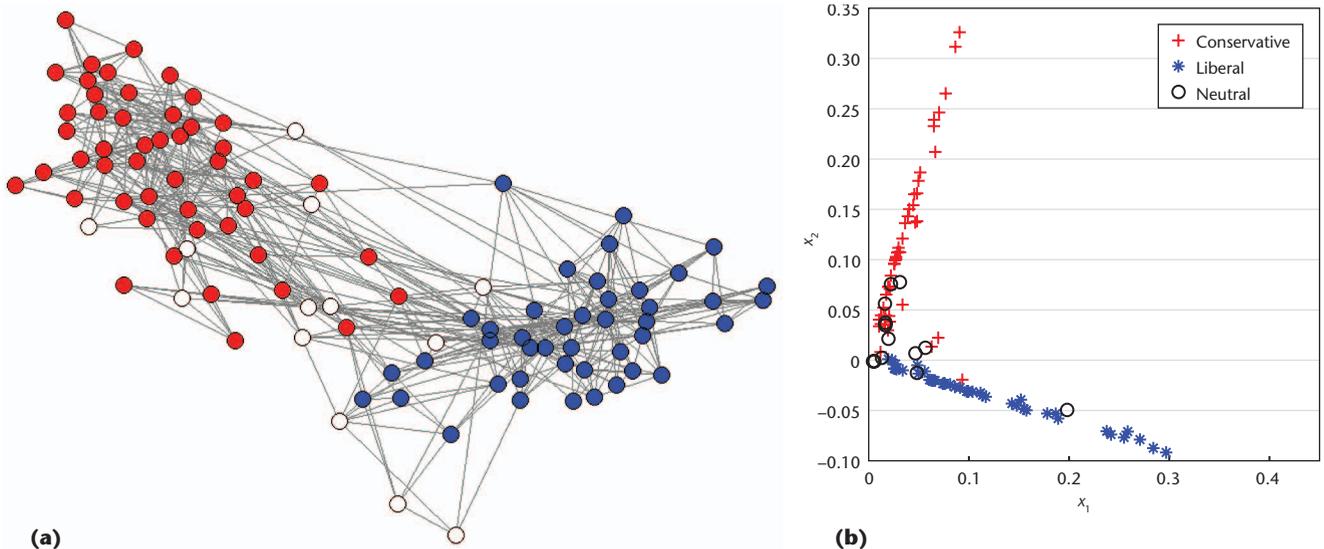


Figure 1. A demonstration of the quasi-orthogonal property for a political-books network. (a) The network.<sup>2</sup> (b) The spectral space.  $x_1$  and  $x_2$  are the first two dimensions of the spectral space. The quasi-orthogonal lines in the spectral space reveal two communities.

of  $u$  in the first  $k$ -dimensional spectral space are the row vector  $(x_{1u}, x_{2u}, \dots, x_{ku})$ .

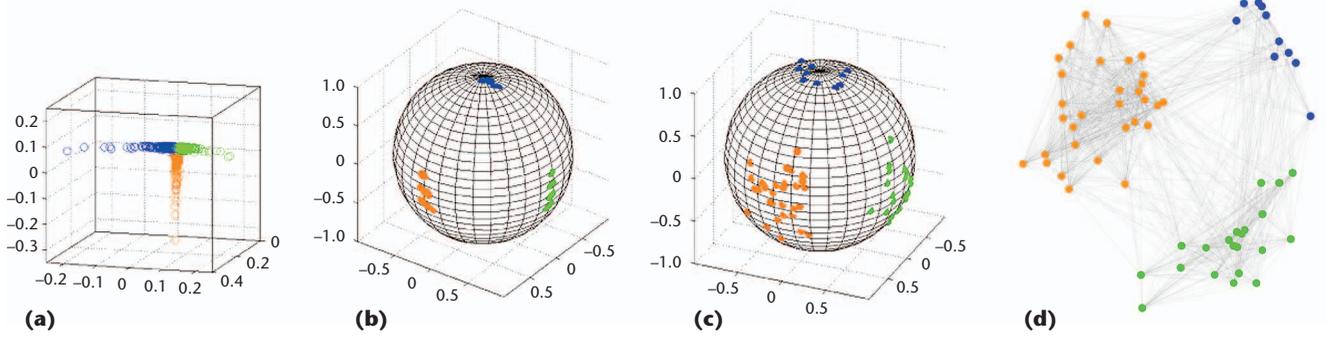
Then, we project the nodes onto a  $k$ -dimensional sphere whose center is at the spectral space's origin:

$$x'_{iu} = \frac{x_{iu}}{\sqrt{\sum_{j=1}^k x_{ju}^2}} \times S_{radius}, \quad 1 \leq i \leq k,$$

where  $S_{radius}$  is the sphere's radius. In all the results of this article, we use 1 for  $S_{radius}$  because it doesn't affect the proportions of distances on the sphere

surface. This projection preserves the topology's main structure and maximizes the distance between communities.

Figure 3 shows the projection results for synthetic datasets containing from three to six communities. In each community, the nodes are fully connected, and we've added 10 percent extra edges to connect nodes from different communities. In all these results,  $k$  is equal to the number of communities. As the circles show, the algorithm clearly reveals the community structures.



**Figure 2.** Our network layout approach. (a) The quasi-orthogonal property of the node distribution in the spectral space. (b) In the spectral space, we project nodes onto the surface of a  $k$ -dimensional sphere. (c) Node dispersion reduces node overlap. (d) Sphere warping transfers the nodes to a 2D space for visualization and interaction. In the final layout, nodes near a community's center are often more important to the network because they have more connections in the community than other nodes.

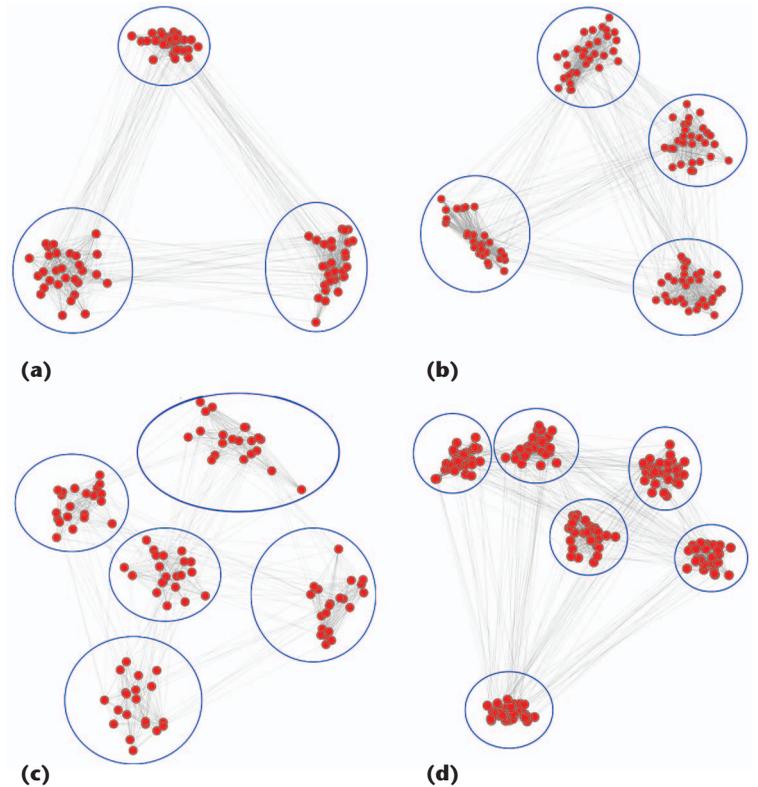
**Node dispersion.** After projecting all the network nodes onto the  $k$ -dimensional sphere, we might have communities with densely connected clustered nodes (see Figure 2b). Generally, users prefer to spread the nodes in the network layout so that they not only can visualize the network's main topology structure but also can analyze individual nodes' and edges' connections. So, we disperse nodes around the sphere's surface.

We base dispersion on the *distance property* of node coordinates in the spectral space.<sup>2</sup> That is, the farther a node is from the spectral space's origin, the greater its importance to its community. According to this property, nodes closer to the origin are more likely to be random nodes that obviously don't belong to any communities. Nodes farther from the origin are often crucial to their communities.<sup>2</sup> So, we preserve the locations of nodes far from the origin and disperse all nodes on the sphere surface according to their distances from the origin.

We disperse the nodes around their projection locations on the sphere instead of their original coordinates. Just a few movements around the original coordinates can greatly change the projection locations, especially when a node is close to the origin. So, we move each node randomly around the sphere surface according to its distance from the origin. In this way, every community's center, represented by the intersection of the corresponding quasi-orthogonal line and the  $k$ -dimensional sphere, remains at the same location. Because important nodes with large connections are distributed near the sphere's surface, they're close to their projection locations. Conversely, random nodes, distributed nearer the spectral space's origin, are more affected by noise.

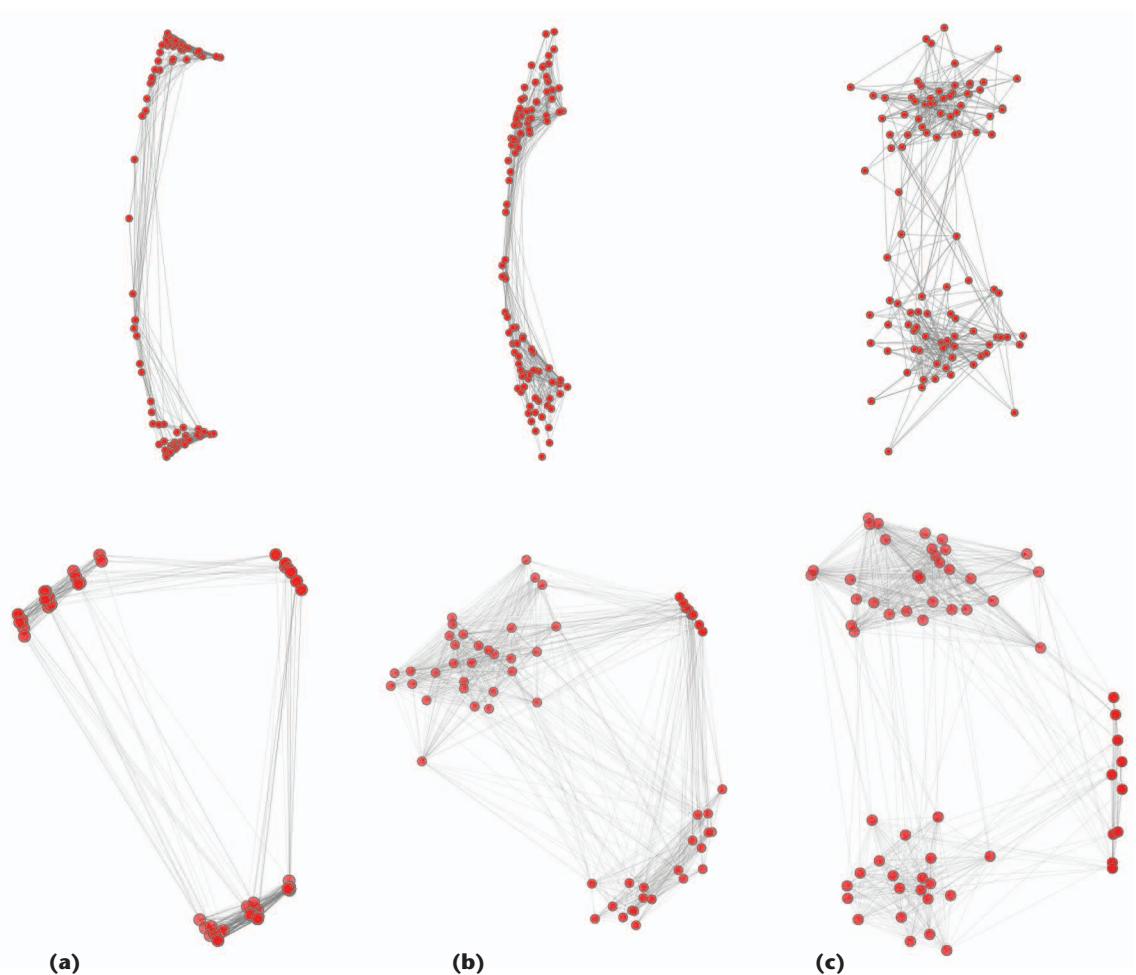
So, we design the dispersion amount to be the inverse of a node's distance from the spectral space's origin:

$$x''_{iu} = x'_{iu} + R \times (1.0 - distance_u) \times r, 1 \leq i \leq k,$$



**Figure 3.** Results of the network layout algorithm for (a) three, (b) four, (c) five, and (d) six communities. As the circles show, the algorithm clearly reveals the community structures.

where  $distance_u$  is the distance of  $u$  to the origin,  $r$  is a random number from  $-1$  to  $1$ , and  $R$  is a parameter to control the degree of dispersion. The selection of  $R$  depends on the number of nodes in the network. Generally, we use a small value for large networks to prevent nodes that were previously separated from different communities from interweaving on the sphere after dispersion. Although moving the nodes on the sphere surface would be ideal, the computation would involve calculating the angles for each of the  $k$  dimensions, which would be more time-consuming. Our



**Figure 4.** Network visualizations (a) without node dispersion, (b) with node dispersion in the  $k$ -dimensional space, and (c) with node dispersion in  $k + 1$  dimensions. The top images involve the political-book dataset; the bottom images involve a synthetic dataset (three communities, 60 nodes, and 871 edges). Dispersion in  $k + 1$  dimensions produces the best results.

approach is computationally efficient and achieves similar effects.

As Figure 4 shows, dispersing nodes on networks with dense nodes is sometimes still difficult. Figure 4b shows that the nodes spread out mainly in one dimension, instead of the entire 2D space. So, we extend the  $k$ -dimensional coordinates to  $k + 1$  dimensions with  $x_{(k+1)u} = 0$ , and we perform dispersion. This step successfully disperses the nodes on the 2D space (see Figure 4c).

**Sphere warping.** To generate the final layout, we warp the  $k$ -dimensional sphere’s surface to a 2D space. First, we collect a distance matrix, which measures node distances on the sphere’s surface. Specifically, for any two nodes  $u$  and  $v$ , we calculate the angle  $\theta$  between the vectors  $\vec{O}u$  and  $\vec{O}v$ , where  $\vec{O}$  represents the spectral space’s origin. Because  $\vec{O}$  is a  $k$ -dimensional zero vector, we calculate  $\theta$  as

$$\theta = \arccos \left( \frac{x''_{iu}x''_{ju}}{|x''_{iu}| \times |x''_{ju}|} \right).$$

Then, we calculate the distance of two nodes on the sphere’s surface:

$$spherical - distance = \frac{\theta \times \pi \times S_{radius}}{180^\circ}.$$

Because the distances (*spherical - distance*) are linear to  $\theta$ , we can ignore this calculation and achieve the same network layout effects.

In the end, the problem is how to generate the 2D network layout that can best approximate the node pair distances we measure from the sphere surface. Of the different high-dimensional projection methods, we chose multidimensional scaling (MDS) because it optimizes the preservation of the relative node pair distances from the  $k$ -dimensional sphere to the 2D layout. As Figures 3 and 4 show, the final network layout projects nodes in the same community onto similar locations on the  $k$ -dimensional sphere and groups them nicely.

**Computational Complexity and Scalability**

Our algorithm’s computational complexity is  $O(N^2)$

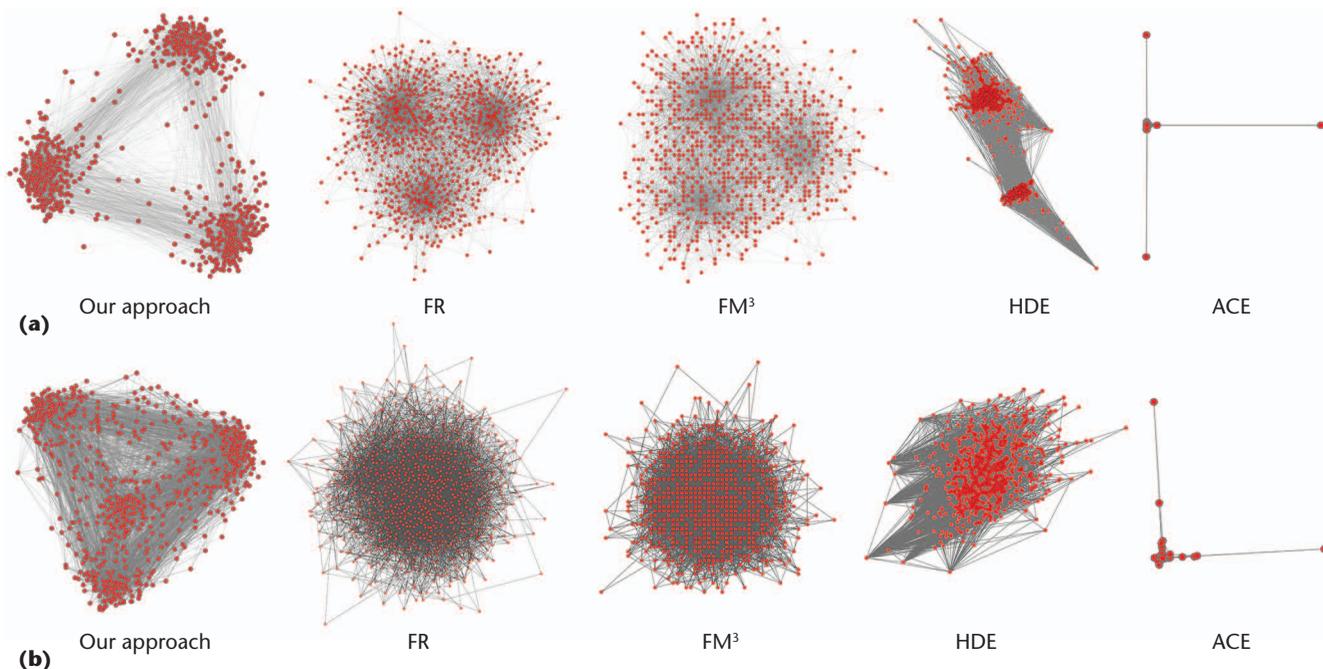


Figure 5. Comparing our approach to the Fruchterman-Reingold algorithm (FR), the fast multipole multilevel method (FM<sup>3</sup>), high-dimensional embedding (HDE), and Algebraic Multigrid Computation of Eigenvectors (ACE). (a) Visualizations of a dataset including three communities with 300, 250, and 230 nodes, respectively. The number of edges between the communities is approximately 20 percent of the number of edges in all the communities. (b) Visualizations of a dataset containing four communities with 200, 180, 170, and 290 nodes, respectively. The number of edges between the communities is approximately 72 percent of the number of edges in all the communities. Our approach achieves the best results by correctly separating the communities and clearly visualizing the topology structures.

(where  $N$  is the number of nodes), which is relatively fast. Spectral decomposition's complexity is also  $O(N^2)$ ; it employs the QR algorithm by a reduction to the Hessenberg form.<sup>5</sup> Node projection and dispersion are only linear to  $N$  and can be easily performed and accelerated. Space warping's complexity is  $O(N^2)$ ; as we mentioned before, it uses MDS, and it follows a hybrid approach.<sup>6</sup>

Our algorithm's scalability is determined mainly by spectral decomposition and MDS. Both have been shown to handle large-scale data in such areas as math, data mining, graph drawing, information visualization, and bioinformatics. Researchers are working on accelerating them, which will help further improve our algorithm's performance.

### Comparing Approaches

Using two datasets, we compared our algorithm with

- the Fruchterman-Reingold algorithm (FR),<sup>7</sup>
- the fast multipole multilevel method (FM<sup>3</sup>),<sup>8</sup>
- high-dimensional embedding (HDE),<sup>9</sup> and
- Algebraic Multigrid Computation of Eigenvectors (ACE).<sup>10</sup>

To generate the FR, FM<sup>3</sup>, and HDE layouts, we used the Tulip graph visualization system (<http://tulip.labri.fr/TulipDrupal>).

To generate the ACE layouts, we used Yehuda Koren's original code, provided by Daniel Archambault. Figure 5 shows the results.

**FR.** This classic force-directed approach has been widely used in network visualization. It simulates the network as a physical system in which nodes are pulled closer together or pushed farther apart by the assigned edges.

As Figure 5a shows, FR visualized the topology of three communities with just 100 iterations for the first dataset. Our approach (see Figure 5a) achieved a clearer visualization of the community structure, with the communities well separated. We also tried FR with other numbers of iterations; the results were almost the same even at 10,000 iterations.

For the second dataset, the community structure was less obvious (the ratio of edges between communities to edges inside communities was higher). FR failed even at 10,000 iterations, whereas our approach still successfully revealed the four communities' topology (see Figure 5b).

Our approach has two advantages over FR. First, it's computationally faster with a complexity of  $O(N^2)$ , whereas FR is  $\Theta(N^2 + E) \times I$ , where  $I$  is the number of iterations. Second, it's noniterative, whereas FR requires users to adjust the number

of iterations, which can affect the network layout results.

**FM<sup>3</sup>.** This accelerated force-directed approach comes with an efficient multilevel scheme and a force model. Its time complexity is  $O(N \log N + E)$ , which is faster than our approach.

Figure 5a demonstrates FM<sup>3</sup> for the first dataset with 30 iterations. The results remained the same with more iterations. Although the results suggest a topology of three communities, the layout isn't as clear as the layouts with our approach and FR.

For the second dataset (10,000 iterations; see Figure 5b), FM<sup>3</sup> only generates a layout with evenly distributed nodes without clear topology structures. This indicates that it doesn't work well for densely connected networks.

**HDE.** This elegant spectral approach is the closest to our approach. It generates graph layouts by embedding information from a network in a high-dimensional space and projecting the nodes to a 2D plane.

HDE works well on networks with mesh-like structures; however, it didn't generate correct layouts for our test datasets. As Figure 5a shows, the bottom group only includes one node, which isn't correct. Figure 5b doesn't show the correct community structure.

**ACE.** Similar to HDE, ACE is a spectral approach that computes node projections in multiple dimensions. It's extremely fast, making it especially suitable for large graphs. (For more on ACE, see the "Related Work in Spectrum-Based Approaches" sidebar.)

As Figure 5 shows, most nodes gather together in one group, with only three nodes outside. Although ACE is the fastest of the five approaches, it didn't work well for our densely connected networks.

## Visualization and Interactive Analysis

A framework of spectrum-based nonrandomness measurements<sup>2</sup> enables visualizations allowing effective interactive analysis of network topologies.

### Visualization

The nonrandomness framework lets us visualize a node's or link's importance to the network topology. For example, an individual's social network tends to consist of members of the same ethnic group, race, or social class. Intuitively, two friends of a given individual are more likely to be friends with each other than they are with other randomly chosen members. The nonrandomness framework quantifies all graph nonrandomness

measures mathematically from the spectra of the network's adjacency matrix. The framework begins by determining an edge's nonrandomness on the basis of the spectral coordinates of the two nodes connected to that edge. A node's nonrandomness is the sum of the nonrandomness values of all the edges connected to it.

We denote the spectral coordinate of  $u$  as  $\alpha_u = (x_{1u}, x_{2u}, \dots, x_{ku}) \in \mathbb{R}^k$  and the spectral coordinate of  $v$  as  $\alpha_v = (x_{1v}, x_{2v}, \dots, x_{kv}) \in \mathbb{R}^k$ . We're interested in two properties:

- The *edge nonrandomness property* is  $R(u,v) = \alpha_u \alpha_v^T = \sum_{i=1}^k x_{iu} x_{iv}$ . We then have  $R(u,v) = \|\alpha_u\|_2 \|\alpha_v\|_2 \cos(\alpha_u, \alpha_v)$ .
- The *node nonrandomness property* is  $R(u) = \sum_{v \in \Gamma(u)} R(u,v)$ , where  $\Gamma(u)$  denotes the neighbor set of  $u$ . We then have  $R(u) = \sum_{i=1}^k \lambda_i x_{iu}^2 = \alpha_u \Lambda_k \alpha_u^T$ , where  $\Lambda_k = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_k\}$ . This means the nonrandomness of  $u$  is the length of its spectral vector with the eigenvalue weighted on corresponding dimensions.

We can use the properties of nodes and edges in the spectral space to reveal important network topology. For example, we can color the nodes according to their distances from the spectral space's origin. With the HSV (hue, saturation, value) color model, we adjust the hue channel with normalized distances and set the other two channels to be full for the most visible effect. As Figure 6a shows, this coloring scheme automatically visualizes the three communities differently. This effect occurs when the communities are different distances from the origin.

Similarly, we design our network visualization on the basis of the distance and nonrandomness properties to emphasize the important network nodes and edges. First, we set a node's transparency to be linear to its distance from the origin (see Figures 6b and 6c). Second, we set a node's size to be linear to its nonrandomness value (see Figure 6b). We normalize all the node nonrandomness values before the visualization. Third, we generally preserve the node color to visualize additional attributes of the network data. As Figures 6b, 6c, and 6d show, we just use one color for all the nodes. Finally, we set an edge's transparency to change linearly with its nonrandomness value. We normalize all the edge nonrandomness values (from 0 to 0.5) before the visualization. As Figures 6c and 6d show, the important edges are often in each community.

### Interactive Analysis

To analyze network topology and the roles of in-

dividual nodes and edges, users can filter or select network data. They can freely combine these methods during interactive exploration, which is often necessary for understanding complex networks.

**Filtering.** For large networks, the main topology structure is often hidden in the visualization of all the nodes and edges. Filtering by various attributes (not just data attributes) lets users remove irrelevant nodes and edges to visualize that structure.

According to the nonrandomness properties, nodes and edges with low nonrandomness values aren't obviously related to any major communities in the network. So, we can use the nonrandomness measurements to filter nodes and edges. This method's main benefit is that users can select nodes along the main topology structure, ranging from visualizing all the network nodes to visualizing only each community's kernel nodes (see Figures 7a through 7c). Users can apply the same selection process to reveal important edges (see Figures 7d through 7f).

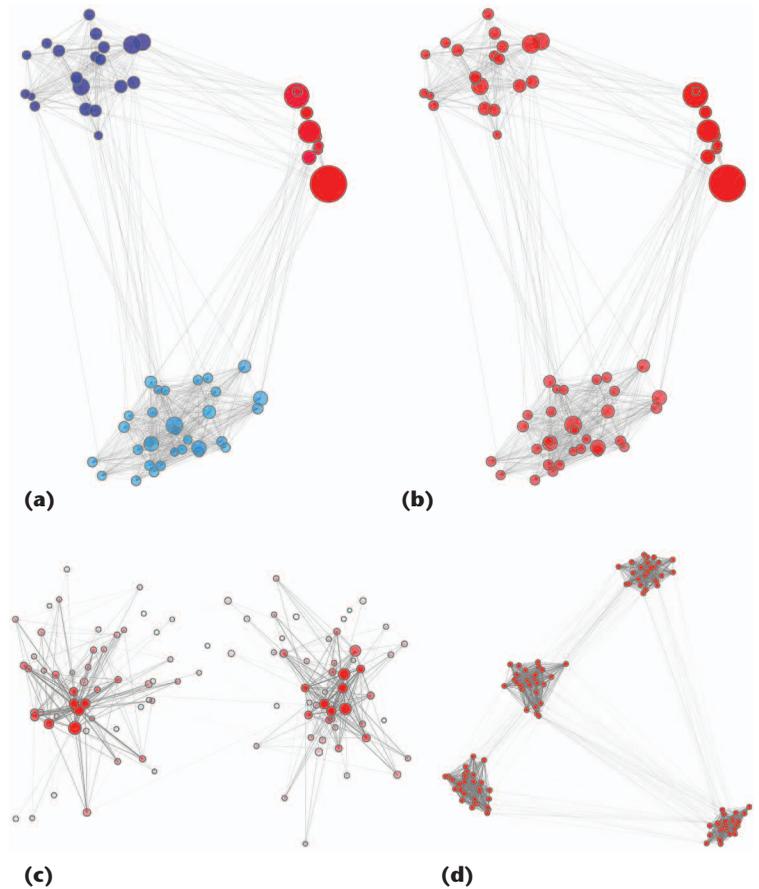
**Selecting examples.** Users can also adjust the network layout on the basis of their knowledge. This lets them identify communities by selecting representative nodes and then simplify a complex network by removing the known communities. This process can repeat until the network topology is completely unraveled. Generally, with a new network, we start from  $k = 2$  and increase the value gradually. Users can also identify a community's representative node by selecting the node with the most connections in the community.

During and after interactive analysis, our layout algorithm visualizes the network using a  $k$  equal to the number of representative nodes. We designed an automatic algorithm to relocate the other nodes surrounding the representative nodes. Assume we have  $m$  representative nodes,  $(v_1, v_2, \dots, v_m)$ . For any other node  $u$ , we first determine which of the  $m$  communities  $u$  belongs to, employing a clustering algorithm. Here, we use  $k$ -means clustering because it's widely used and lets us freely define the number of communities or clusters.

Second, we shift  $u$  from its original location in the spectral space:

$$\bar{u}' = \bar{u} + p \times \frac{\bar{u}\bar{v}_i}{\|\bar{u}\bar{v}_i\|},$$

where  $p$  is a parameter controlling the amount of movement. The result is that the rest of the nodes in each community are close to the community's representative node.



**Figure 6. Network visualization styles:** (a) distance-based color for a synthetic dataset with three communities, 60 nodes, and 737 edges; (b) node transparency, size, and color for a synthetic dataset with three communities, 60 nodes, and 737 edges; (c) node and edge transparency on a real political-books dataset with 105 nodes and 441 edges; and (d) edge transparency for a synthetic dataset with four communities, 120 nodes, and 2,262 edges. The layouts clearly visualize the community structure and emphasize the crucial nodes with their connections.

**Two case studies.** Here we show how we performed interactive analysis with two real datasets. We rendered the nodes so that all the nodes in a community are the same color.

The first dataset comprised American college football games between Division IA schools during the regular 2000 season. There were 115 teams and 12 communities, including one for independent teams. As Figure 8a shows, initially the network topology could separate several large communities; however, nodes were obviously floating between communities. We first chose tightly grouped communities and representative nodes (see Figures 8a through 8e). Every time, we started with  $k = 2$  and gradually increased  $k$  so that we could choose clearly isolated communities. We stopped at Figure 8e because the topology structure was clear. According to the selected 11 representative nodes, we set  $k = 11$  and generated the final network layout (see Figure 8f).

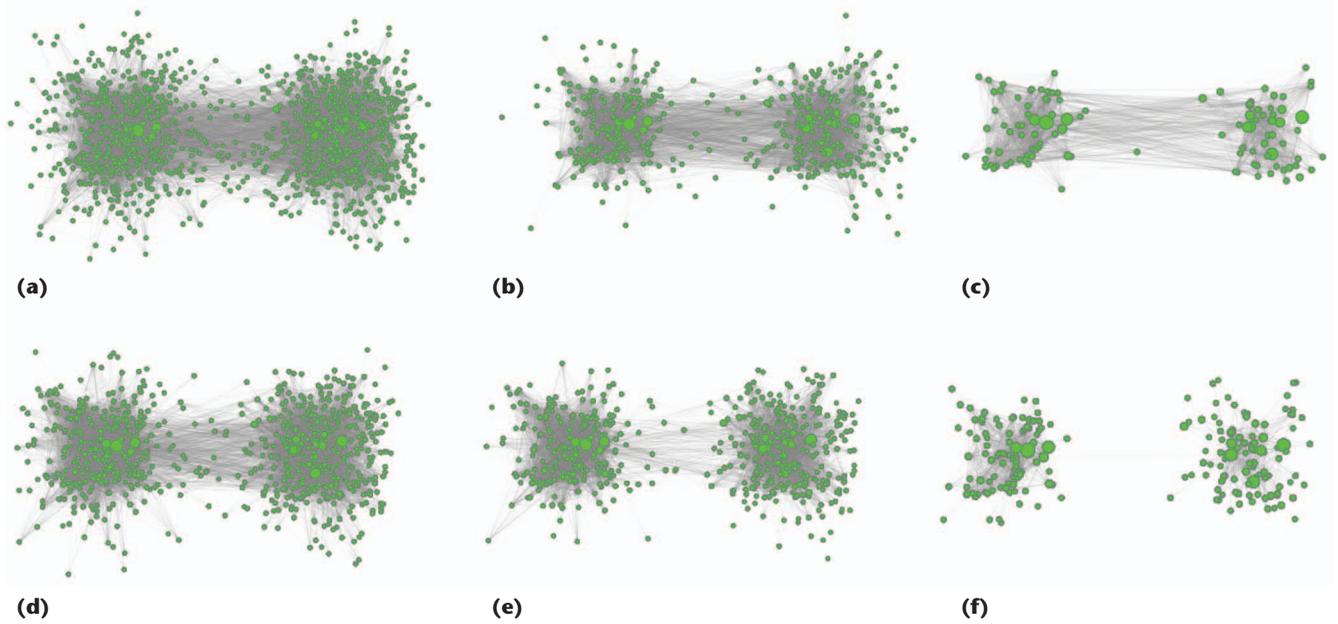


Figure 7. Filtering results on a political-blogs dataset with 1,222 nodes and 33,428 edges: (a) the initial layout, (b) node nonrandomness  $> -0.045$ , (c) node nonrandomness  $> 0.258$ , (d) edge nonrandomness  $> 0.001$ , (e) edge nonrandomness  $> 0.003$ , and (f) edge nonrandomness  $> 0.01$ . The node nonrandomness ranged from  $-0.3495$  to  $5.7343$ ; the edge nonrandomness ranged from  $-0.0016$  to  $0.0314$ . With more node and edge filtering, the kernel topology structure appears clearer.

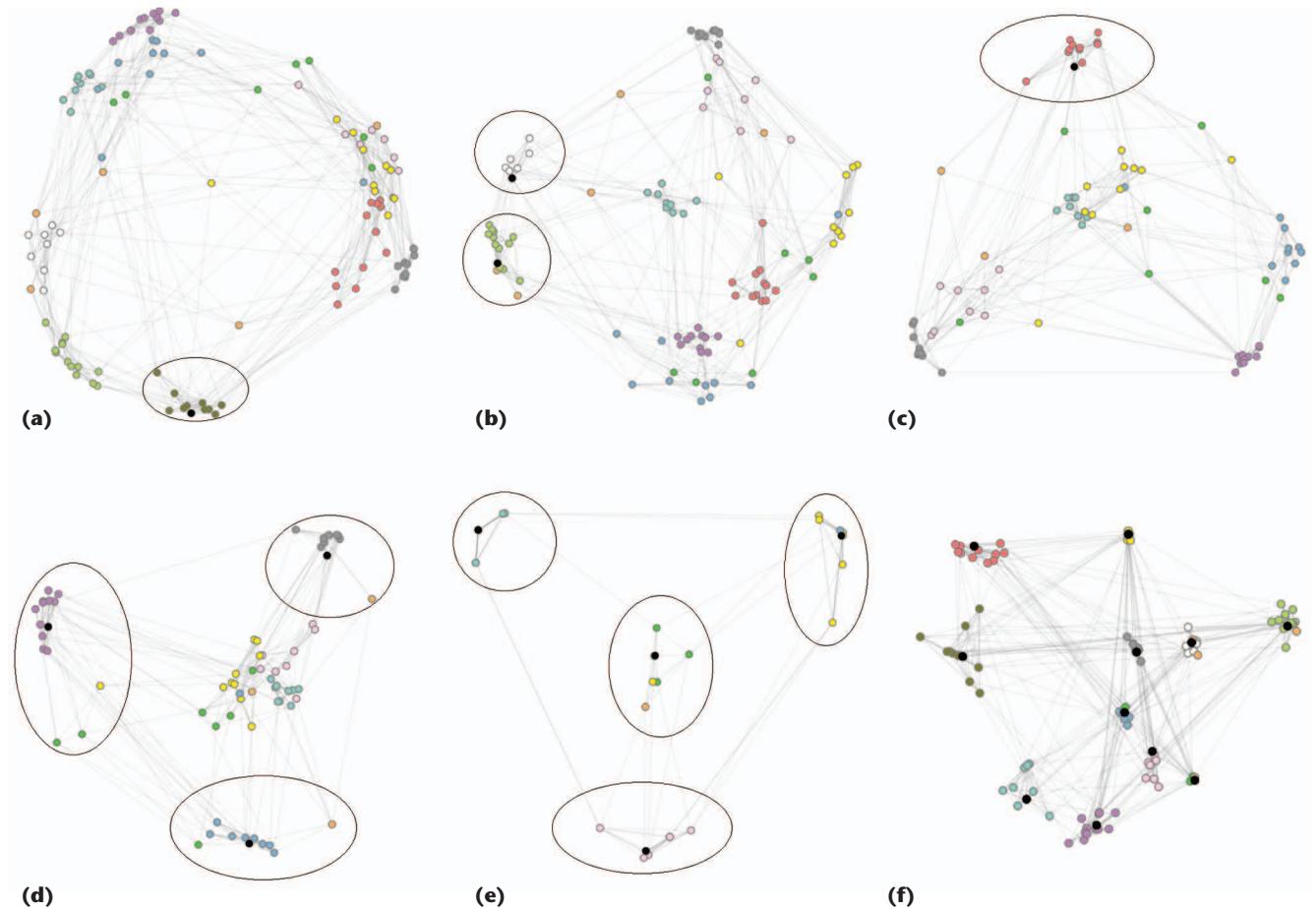


Figure 8. A sequence of interactive analysis of a network of American college football games: (a)  $k = 3$ , (b)  $k = 5$ , (c)  $k = 5$ , (d)  $k = 7$ , (e)  $k = 4$ , and (f) the final layout. Each selected community is circled separately; representative nodes are black. The final results closely match the original datasets' topology structure.

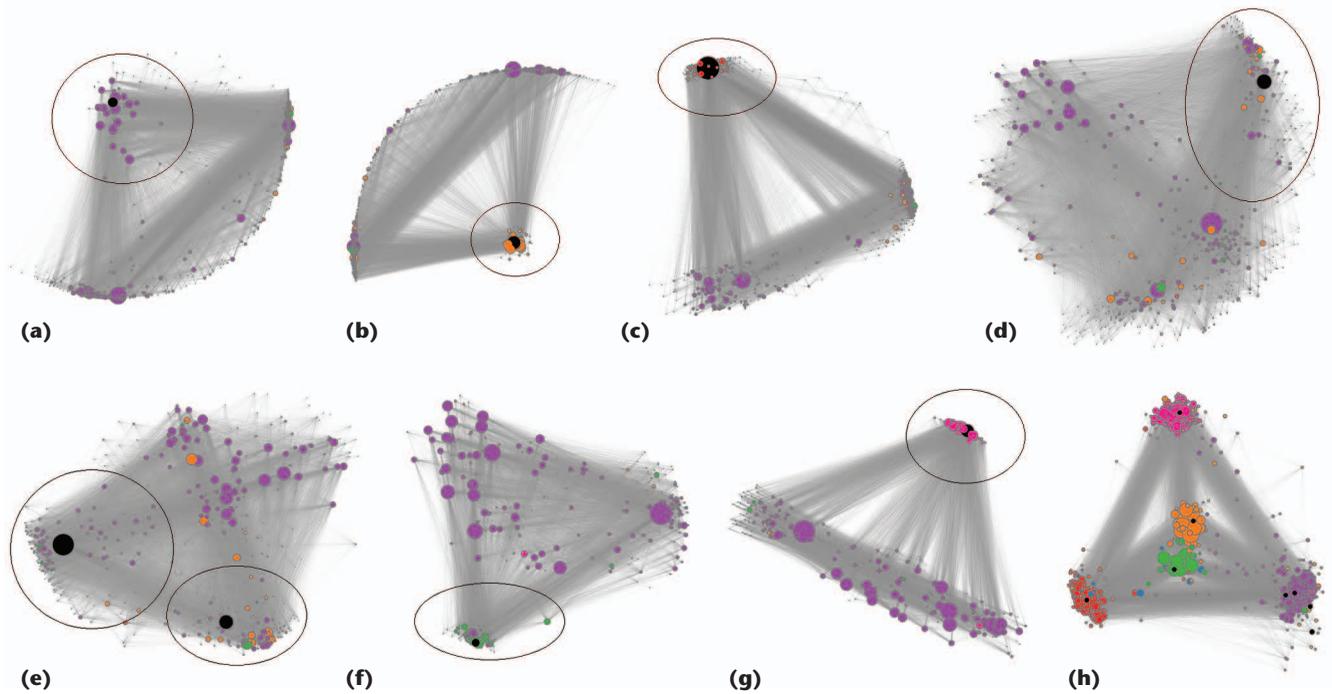


Figure 9. A sequence of interactive analysis of a CiteSeer dataset of scientific publications: (a)  $k = 3$ , (b)  $k = 3$ , (c)  $k = 3$ , (d)  $k = 6$ , (e)  $k = 5$ , (f)  $k = 5$ , (g)  $k = 3$ , and (h) the final layout. Each selected community is circled separately; representative nodes are black. The final results closely match the original datasets' topology structure.

Comparing our result with the original network topology, we can tell that our approach successfully identified the major topology structures. The main difference is due to the independent schools, rendered in orange. The teams in this community had more connections with the other conferences than with their own community. The teams in the other 11 communities were all categorized correctly.

The second case study used a CiteSeer dataset with 3,312 scientific publications from six research communities and 4,536 edges.<sup>11</sup> Figure 9 demonstrates the selection of groups and representative nodes. To visualize important topology structure, we filtered nodes and edges. Node size increased according to a node's distance from the origin.

Because the information retrieval community (purple) was the largest (containing 30 percent more connections than any other group), it dominated the initial networks. We selected four representative nodes. This didn't affect the final layout (see Figure 9h), in which those four nodes are closely located and this group is still clearly separated.

The final layout has five communities: information retrieval, agents (red), databases (green), machine learning (orange), and human-computer interaction (pink). The only missing community, artificial intelligence (blue), overlaps with the agents group and is dragged toward the databases and machine-learning communities. This is because there are twice as many connections between

the agents and artificial intelligence communities as there are within the artificial intelligence community. Some artificial intelligence nodes are close to the machine-learning and databases communities because the artificial intelligence community has more connections with these two communities than with the human-computer interaction and information retrieval communities.

Interactive analysis can tolerate errors in the selection process. For example, all three communities selected in Figure 8d included other communities' nodes. These errors were corrected in the final result (see Figure 8f), which globally recomputed each node's relationships.

Our data transformation framework differs significantly from previous network layout approaches, which often come directly from the network connection information. It also differs from previous spectrum-based approaches, which often concentrate on showing the data with common interests.

Our approach will also work for networks other than social networks. For example, cybersecurity analysts could identify fraud or attacks because malicious nodes often demonstrate different distribution patterns in the spectral space. We believe that advanced spectrum-based analysis should be explored for developing more effective data transformation and visual-analytics approaches. ■

## References

1. A. Seary and W. Richards, "Spectral Methods for Analyzing and Visualizing Networks: An Introduction," *Nat'l Research Council, Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, Nat'l Academies Press, 2003, pp. 209–228.
2. X. Ying and X. Wu, "On Randomness Measures for Social Networks," *Proc. 9th SIAM Conf. Data Mining*, Soc. for Industrial and Applied Mathematics, 2009, pp. 709–720.
3. V. Krebs, "New Political Patterns," 2008; [www.orgnet.com/divided.html](http://www.orgnet.com/divided.html).
4. M.E.J. Newman, "Modularity and Community Structure in Networks," *Proc. Nat'l Academy of Sciences*, vol. 103, no. 23, 2006, pp. 8577–8582.
5. L.N. Trefethen and D. Bau, *Numerical Linear Algebra*, Soc. for Industrial and Applied Mathematics, 1997.
6. A. Morrison, G. Ross, and M. Chalmers, "Fast Multidimensional Scaling through Sampling, Springs and Interpolation," *Information Visualization*, vol. 2, no. 1, 2003, pp. 68–77.
7. T.M.J. Fruchterman and E.M. Reingold, "Graph Drawing by Force-Directed Placement," *Software: Practice and Experience*, vol. 21, no. 11, 1991, pp. 1129–1164.
8. S. Hachul and M. Jünger, "Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm," *Graph Drawing*, LNCS 3383, Springer, 2004, pp. 285–295.
9. D. Harel and Y. Koren, "Graph Drawing by High-Dimensional Embedding," *Revised Papers from the 10th Int'l Symp. Graph Drawing (GD 02)*, Springer, 2002, pp. 207–219.
10. Y. Koren, L. Carmel, and D. Harel, "Drawing Huge Graphs by Algebraic Multigrid Optimization," *Multi-scale Modeling and Simulation*, vol. 1, no. 4, 2003, pp. 645–673.
11. P. Sen et al., "Collective Classification in Network Data," *AI Magazine*, vol. 29, no. 3, 2008, pp. 93–106.

**Xianlin Hu** is a PhD student in the Department of Computer Science at the University of North Carolina at Charlotte. Her research interests include social network analysis and visualization. Hu received a BS in information technology from the North China University of Technology. Contact her at [amust1987@gmail.com](mailto:amust1987@gmail.com).

**Aidong Lu** is an associate professor in the Department of Computer Science at the University of North Carolina at Charlotte. Her research interests are visualization and human-computer interaction. Lu received a PhD in electrical and computer engineering from Purdue University. Contact her at [aidong.lu@unc.edu](mailto:aidong.lu@unc.edu).

**Xintao Wu** is a professor in the Department of Software and Information Systems at the University of North Carolina at Charlotte. His research interests include data mining and data privacy and security. Wu received a PhD in information technology from George Mason University. He's a member of ACM. Contact him at [xwu@unc.edu](mailto:xwu@unc.edu).

# IEEE VR 2013

20th IEEE Virtual Reality

16-20 March 2013

Orlando, FL, USA

IEEE VR 2013 is the premier international conference and exhibition on virtual reality. Find the brightest minds, the most innovative research, the leading companies and the most stimulating discussions in the fields of virtual environments, augmented reality, haptics, and 3D user interfaces.

*Register today!*

<http://conferences.computer.org/vr/2013>

