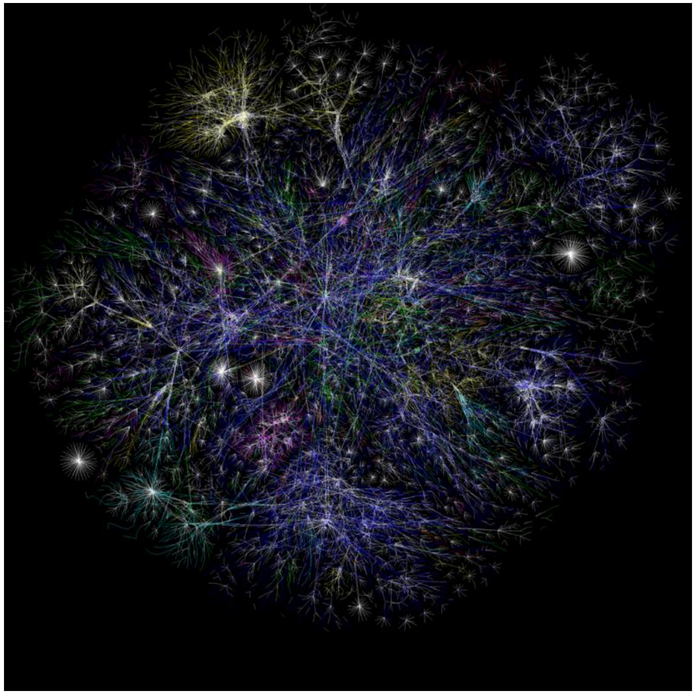
# Networks

COMP8503
Advanced Topics in Visual Analytics

# Network Size

- Size of the WWW: ~40 billion pages (http://www.worldwidewebsize.com/)



[http://opte.org]

# Network size

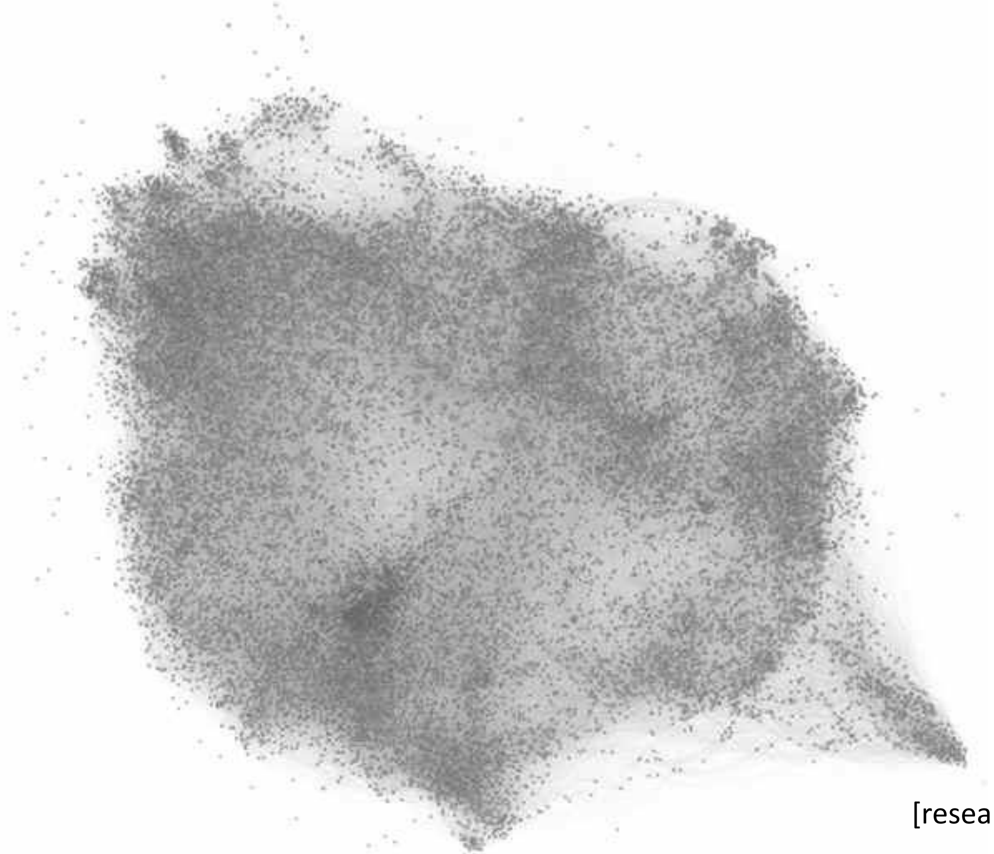
- Facebook: 1.32 billion users
- Twitter: 255 million active users
- WeChat/WeiXin: 438.2 million active users

(Data from:
http://expandedramblings.com/index.php/resource-how-many-people-use-the-top-social-media/)

[Paul Butler, http://fbmap.bitaesthetics.com/]

# Network size

- Citation network: > 250 million articles



[researchtrends.com]

19,562 journals, linked by 377,729 citation relationships

# Graph Drawing

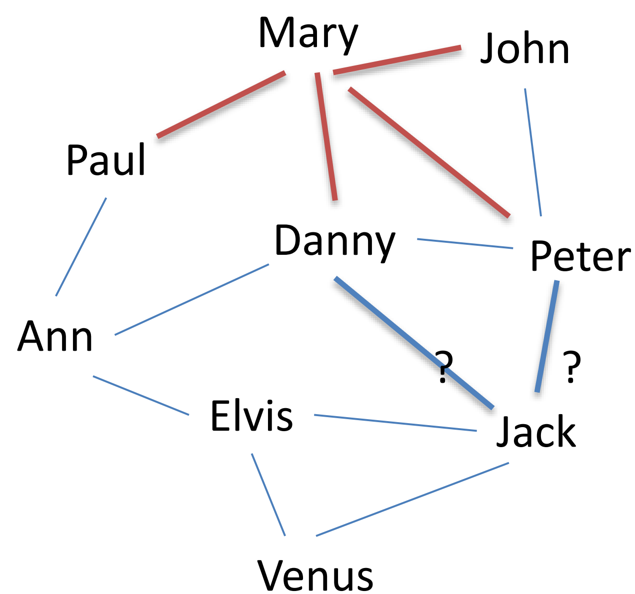
- Direct calculation based on graph structure
  - Spanning tree
  - Adjacency matrix layout
- Optimization-based
  - Optimizing the graph aesthetic constraints
  - Force-directed layout

# Spanning Tree Layout

- Many graphs have tree-like structure or useful **spanning trees** (i.e., trees that include all vertices but only some edges of the original graph)
  - WWW, Social Networks
- To extract a spanning tree from a graph and visualize the tree (which is efficient)
- Drawing of a graph is in general non-deterministic, and a spanning tree layout offers predictability
- Spanning Tree can be obtained by
  - Breadth-First Search (BFS) / Depth-First Search (DFS)
  - Min/max spanning tree

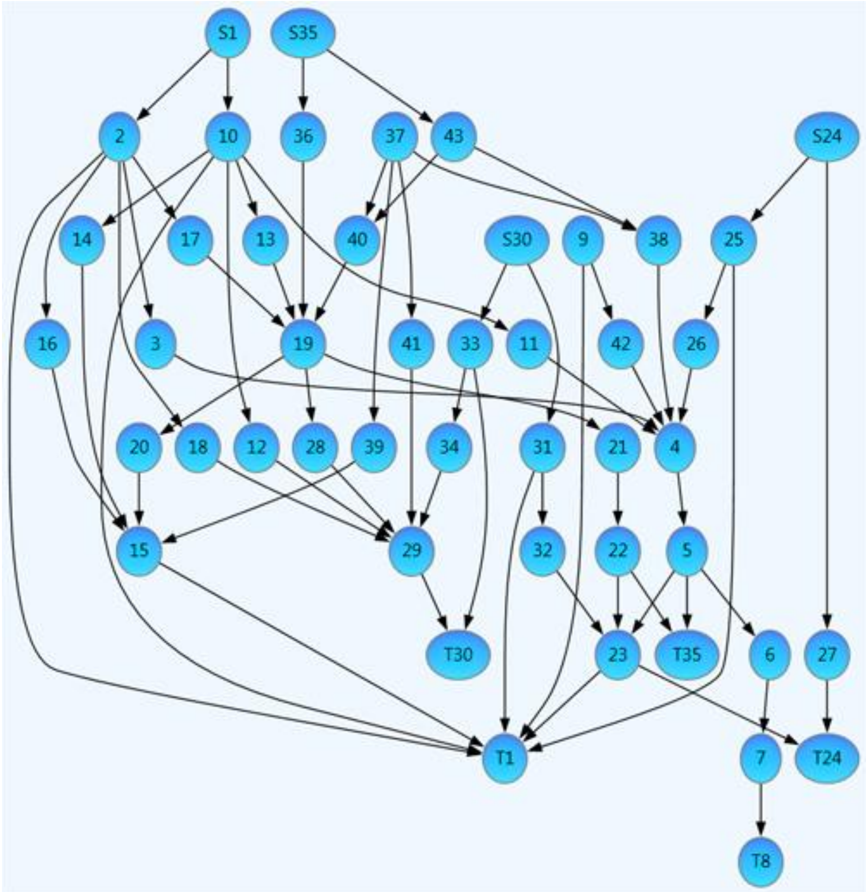
# Spanning Tree Layout

- Which vertex should be the root?
  - A node with minimal distance to all other nodes is a good candidate
- May result in arbitrary parent node



# Sugiyama-style Graph Layout

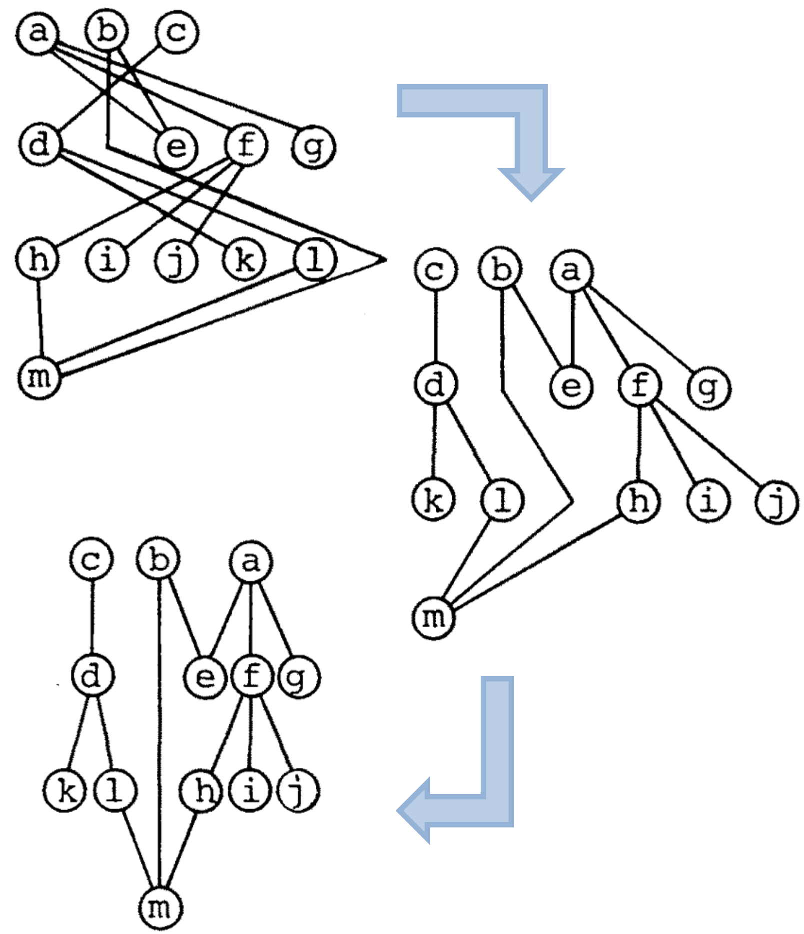
- Hierarchical or layered graph drawing
- Originally for general directed graphs by Sugiyama and his colleagues




[Microsoft Automatic Graph Layout Project]

# Sugiyama-style Graph Layout

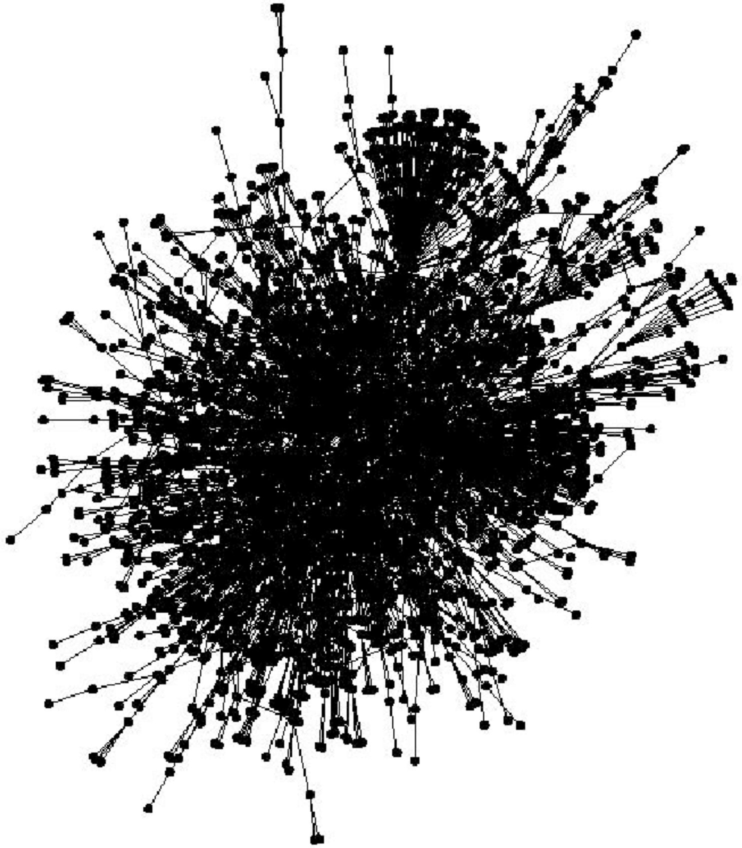
- For general graph layout:
  1. Reverse edges to remove cycles
  2. Assign nodes to layers
     - Dummy nodes are added if an edge spans multiple layers
  3. Order nodes in each layer to minimize edge crossings
  4. Restore edge orientations and remove dummy vertices
  5. Edges are drawn as polylines or spline curves to avoid intersection

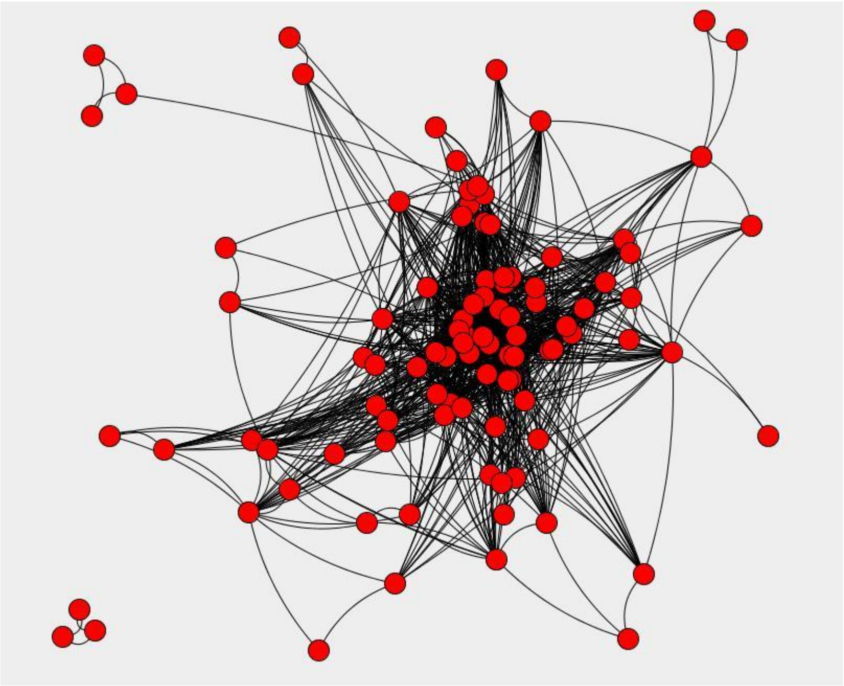



[Sugiyama et al., "Methods for Visual Understanding of Hierarchical System Structures," *IEEE Transactions on Systems, Man and Cybernetics*, 1981.]

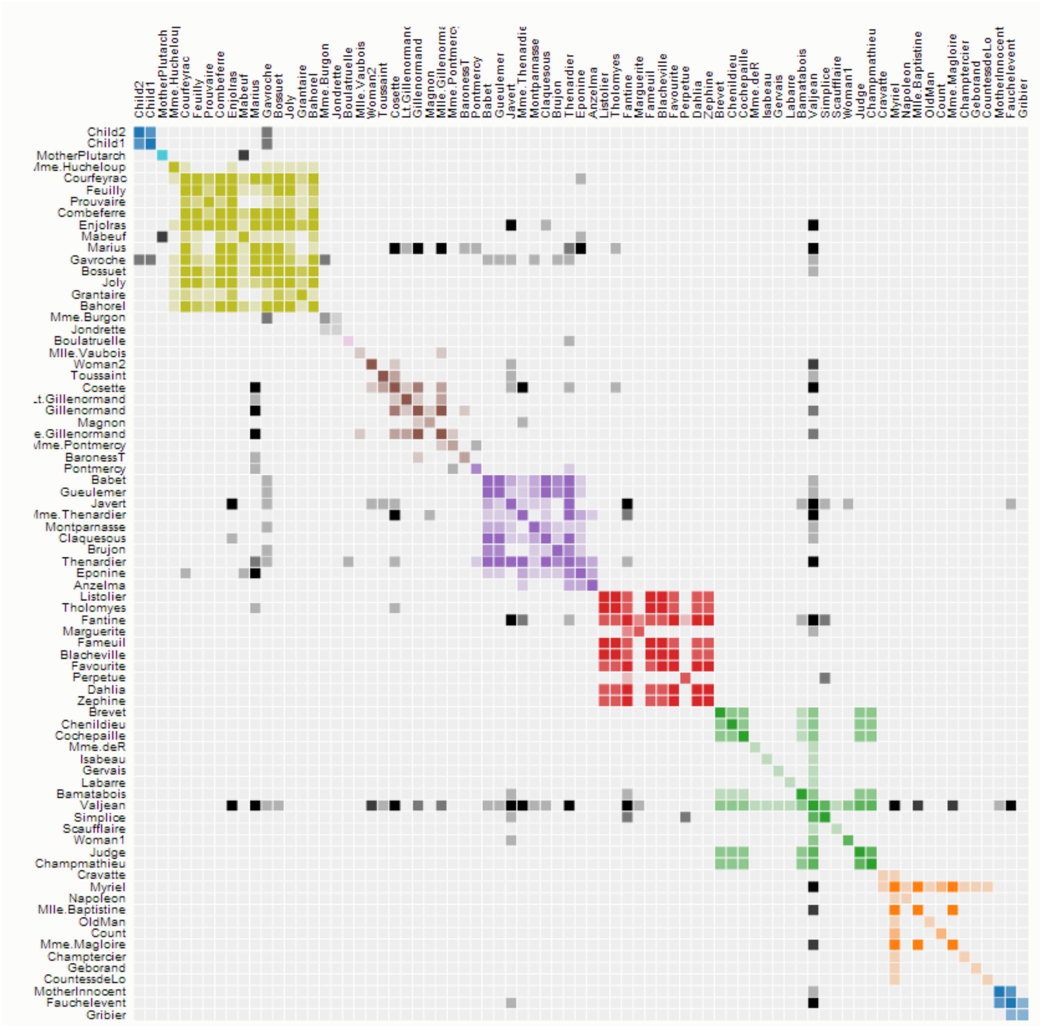
# Node-Link Layout

- Severe edge crossings and cluttering





# Adjacency Matrices




*Les Misérables* Co-occurrence

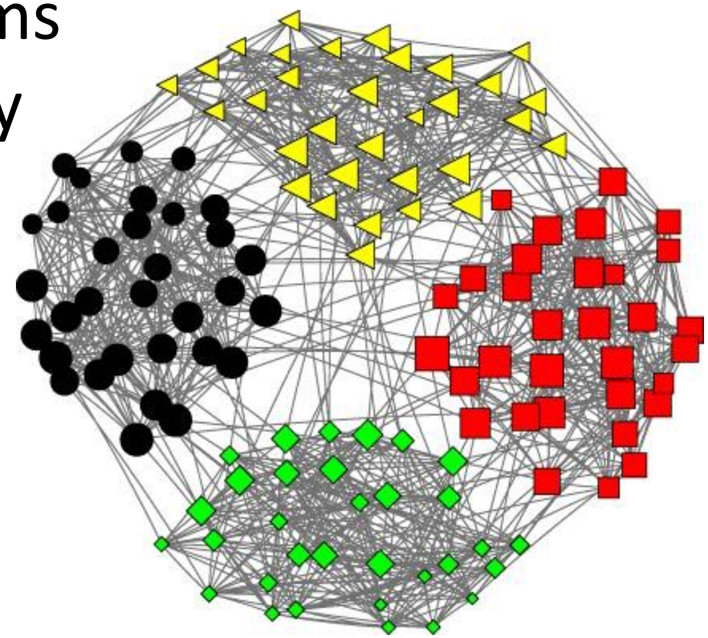
[http://bost.ocks.org/mike/miserables/]

# Clustering

- To reduce the number of visible elements in a graph being viewed
  - Improves clarity
  - Increases performance of layout and rendering
- Structure-based clustering
  - Use only structural information of a graph
- Content-based clustering
  - Use semantic data associated with graph elements
  - Application specific
- Can facilitate filtering (de-emphasize) and search (emphasize) in graph

# Clustering

- A cluster is commonly taken as one with the least number of edges between members
  - or with the minimum total weight of the edges connecting members for graphs with weighted edges
- Force directed layout algorithms can also form clusters naturally



# Edge Bundling

- Clustering of edges instead of nodes
- To reduce cluttering

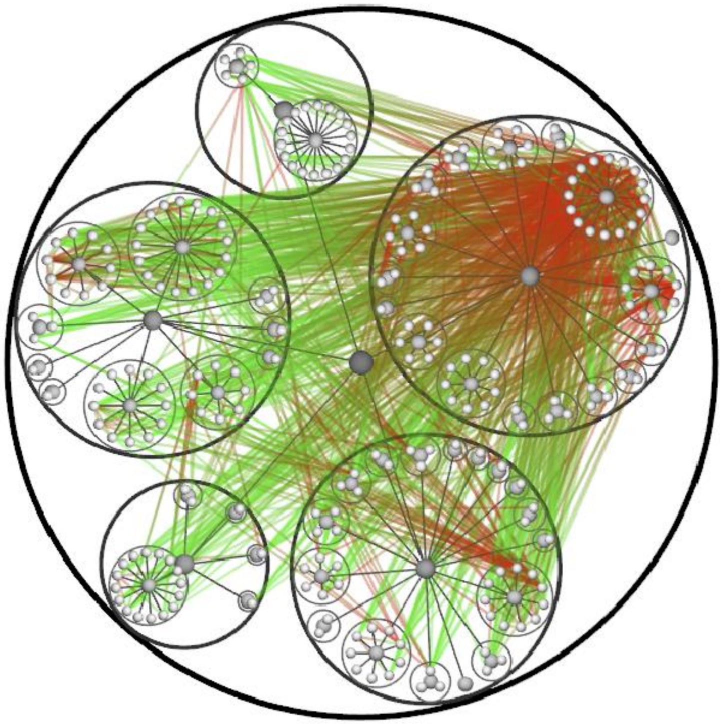
- Examples:
  – Hierarchical Edge Bundles
  – Geometry-Based Edge Clustering

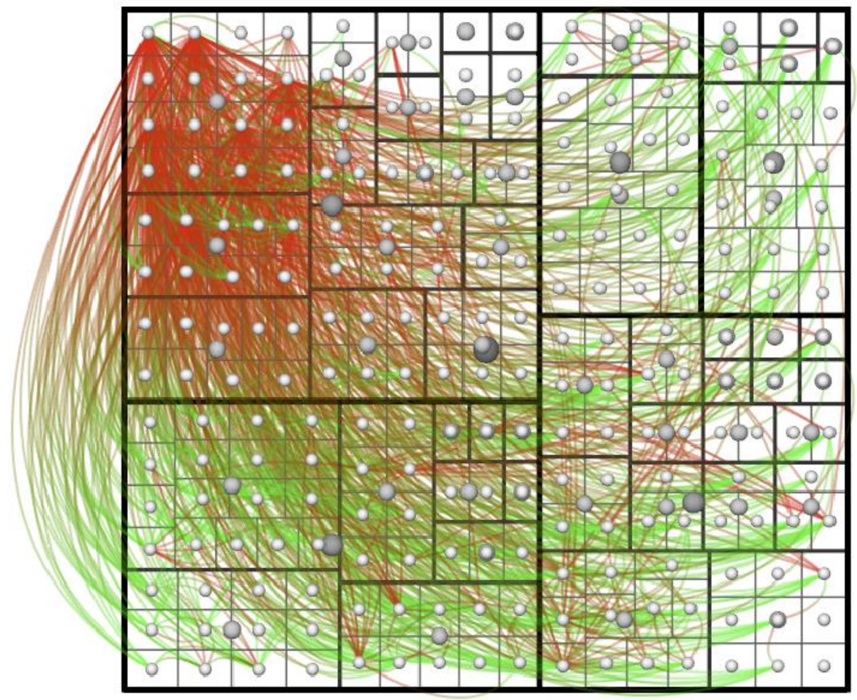
# Hierarchical Edge Bundles

- There are data sets with both hierarchical and non-hierarchical (adjacency) relations.
  - Source codes for a software:
    - Hierarchical: directories -> files -> classes
    - Adjacency: dependencies of classes
  - Social networks:
    - Hierarchical: circles or groups of people -> individuals
    - Adjacency : nature and if people are acquainted
  - Citation networks:
    - Hierarchical: institutions -> departments -> publications
    - Adjacency: citations among publications

# Hierarchical Edge Bundles

- Visualization examples without edge bundling



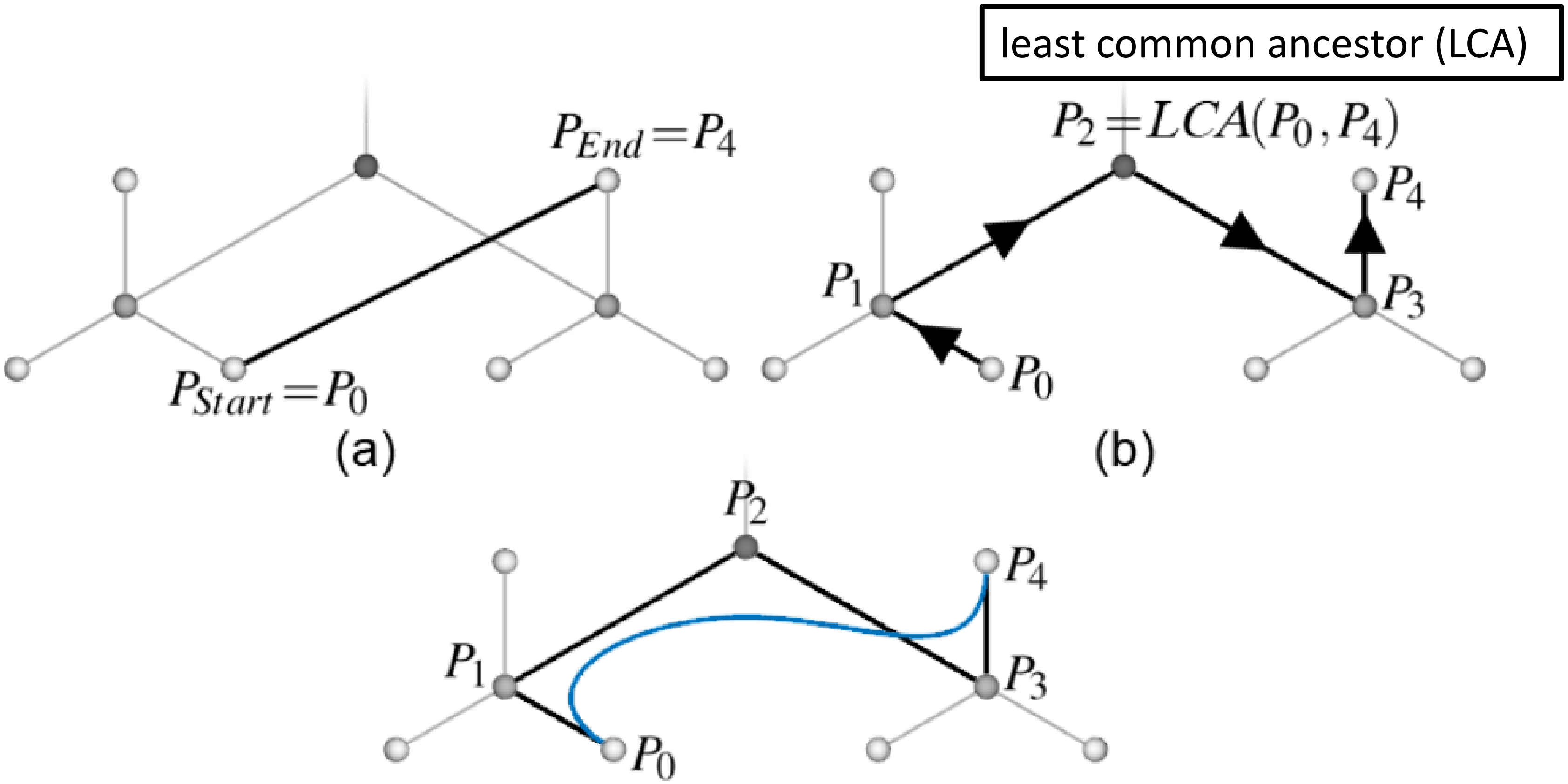


Colored edges representing adjacency relations on (left) balloon trees, and (right) tree maps.

[Holten, "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data," *TVCG* , 2006]

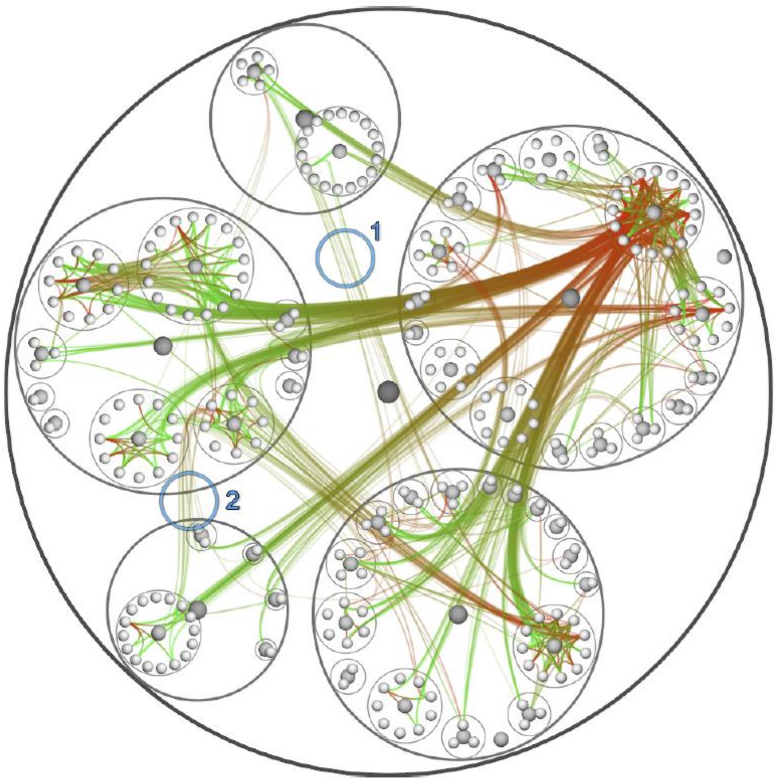
# Hierarchical Edge Bundles

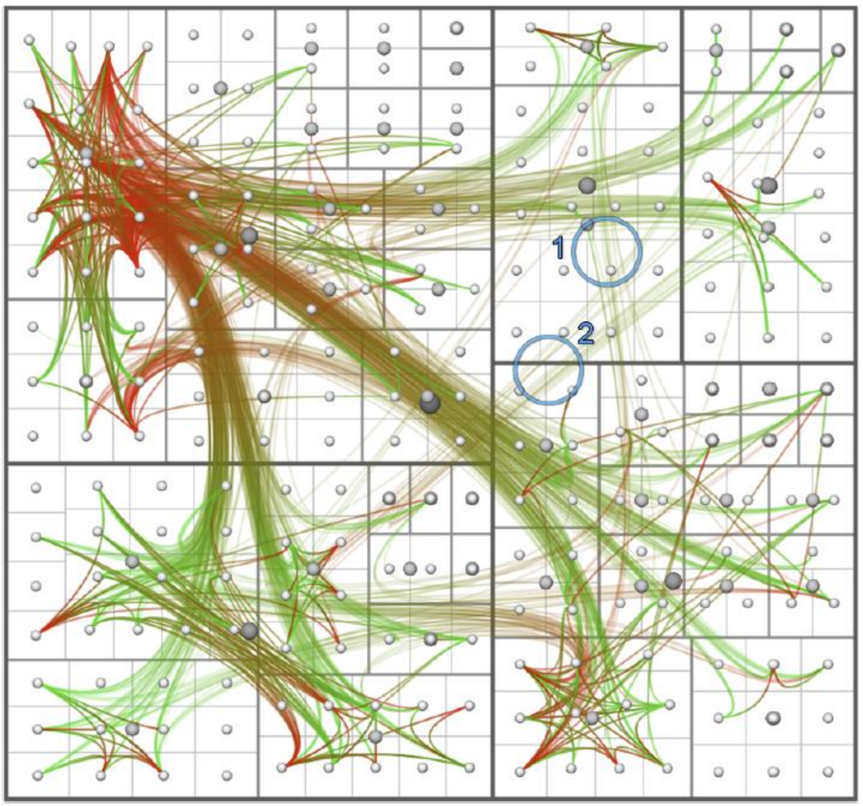
- Bundle adjacency edges along tree hierarchy



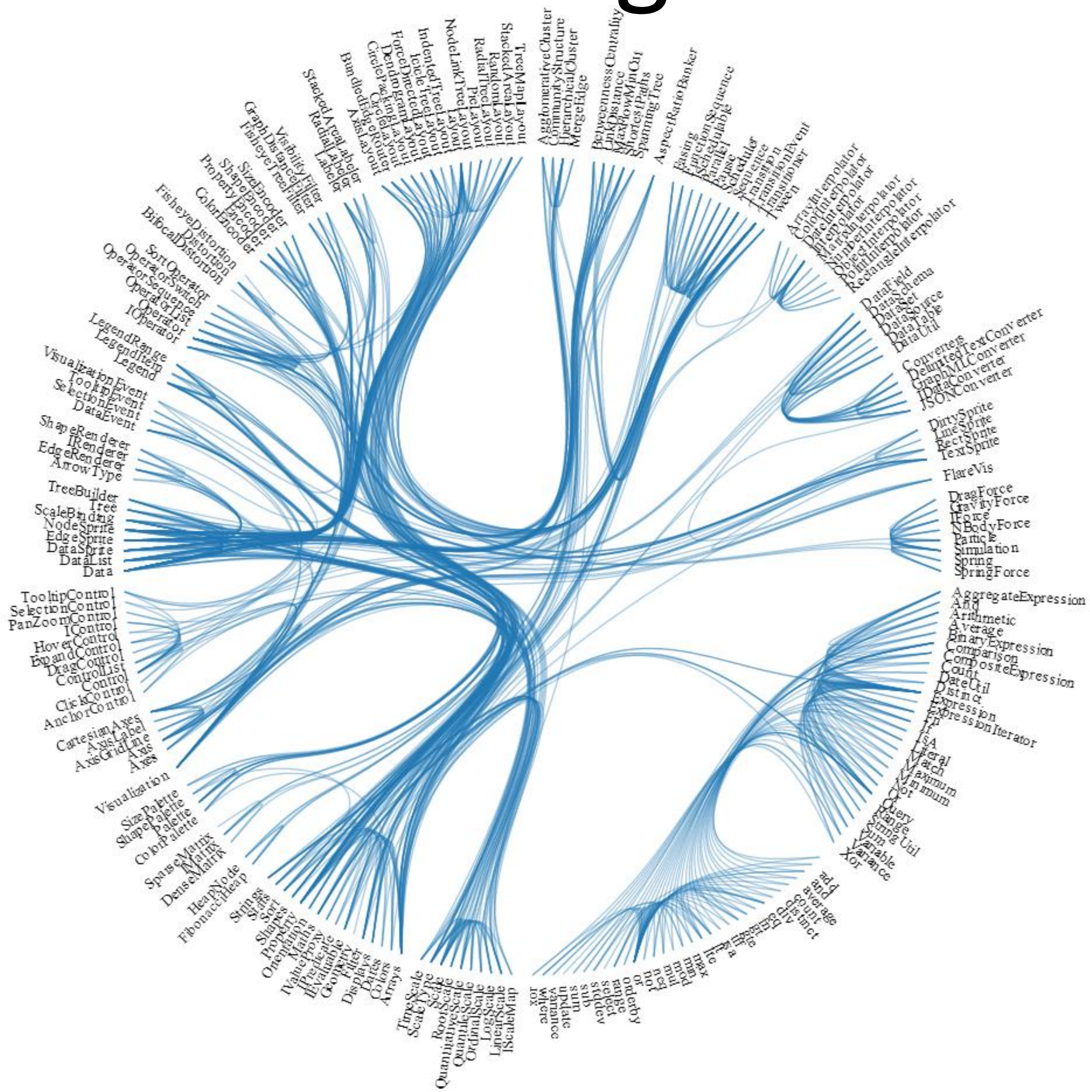
# Hierarchical Edge Bundles

- Examples with hierarchical edge bundling



# Hierarchical Edge Bundles




[http://mbostock.github.io/d3/talk/20111116/bundle.html]

# Geometry-Based Edge Clustering

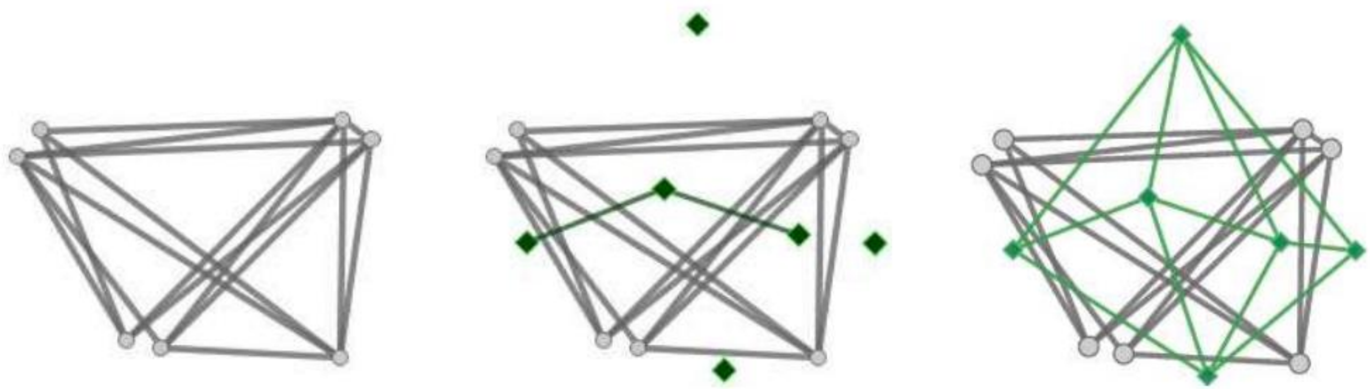
- A control mesh is generated from the graph, based on the underlying graph patterns (node position + edge distribution), to guide the edge clustering process
- A good control mesh can help
  - reduce the number of edge crossings
  - bundle edges with similar directions and lengths
  - minimize the distances between original straight-line edges and resulting polyline edges

[Cui et al., "Geometry-Based Edge Clustering for Graph Visualization," *TVCG,* 2008]

# Geometry-Based Edge Clustering

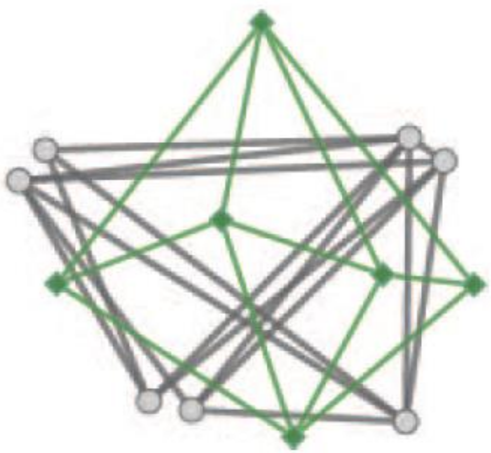
- Edge bundles are formed by forcing all edges to pass through some control points on the mesh


- Control mesh can be manually specified by the user or automatically generated

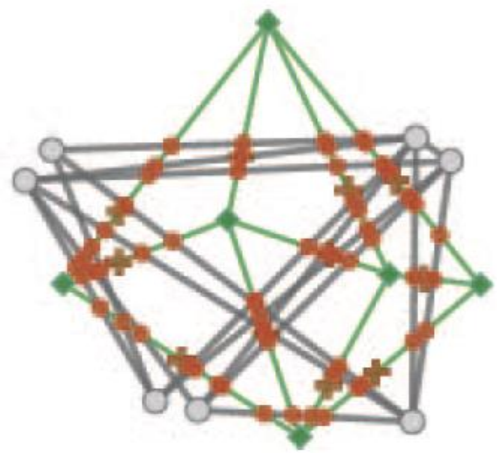


# Geometry-Based Edge Clustering

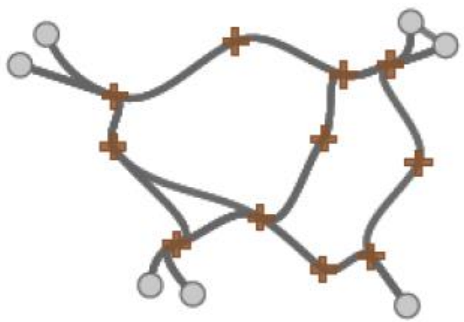
- Edges are then clustered by merging intersection points of the graph edges along the same edge of the control mesh.



A graph with control mesh in green

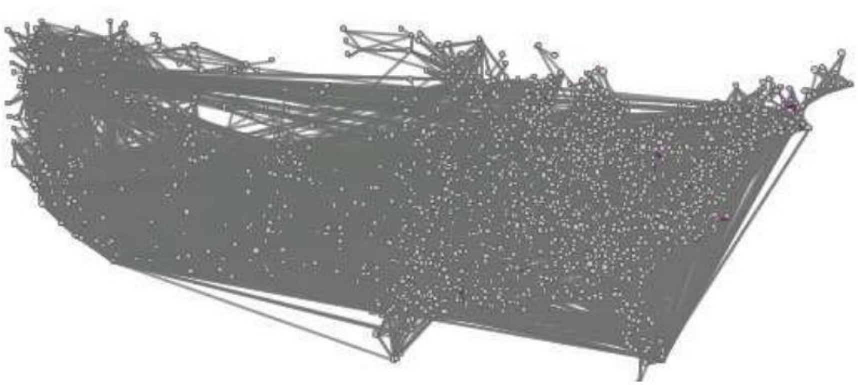


Intersections ( • ) and control points ( + )



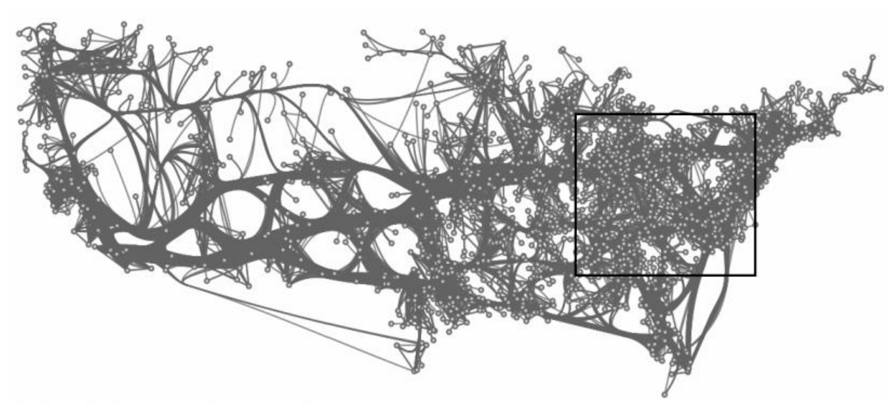
Edges merged

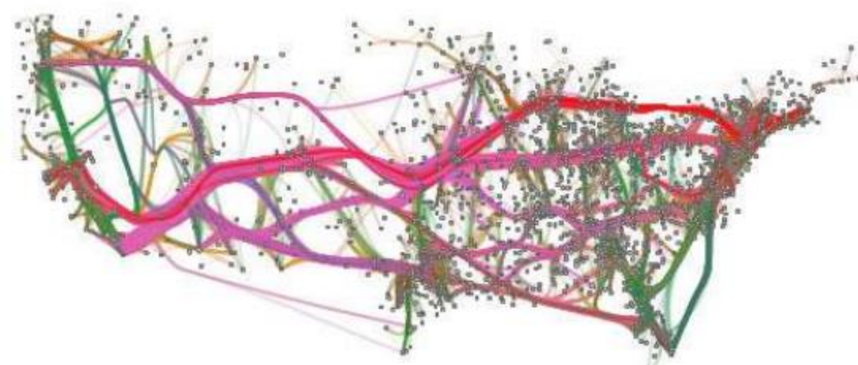
# Geometry-Based Edge Clustering



U.S. immigration graph with
1790 nodes and 9798 edges.

Result of edge clustering





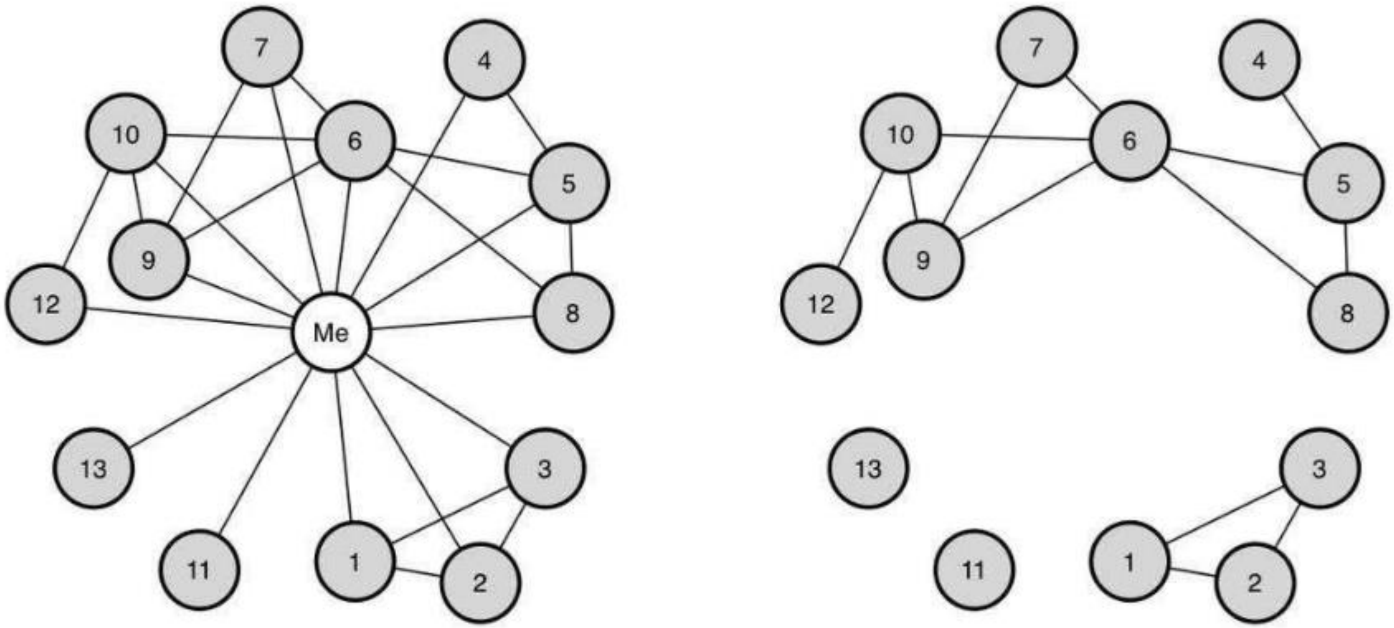
Edge clustering + color + opacity

# Filtering

- Not all edges are needed in a visualization
- Removing some “redundant” edges to better reveal network topology, and to facilitate the clustering process
- E.g., Edges linking you and your friends in your Facebook network

# Filtering

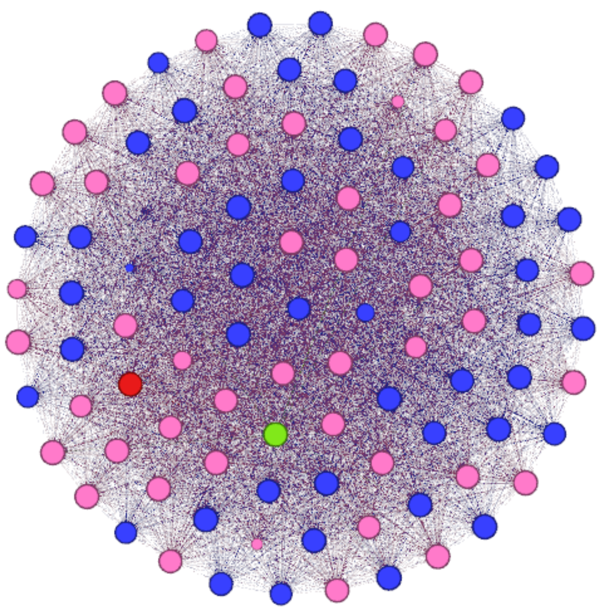
- Ego network



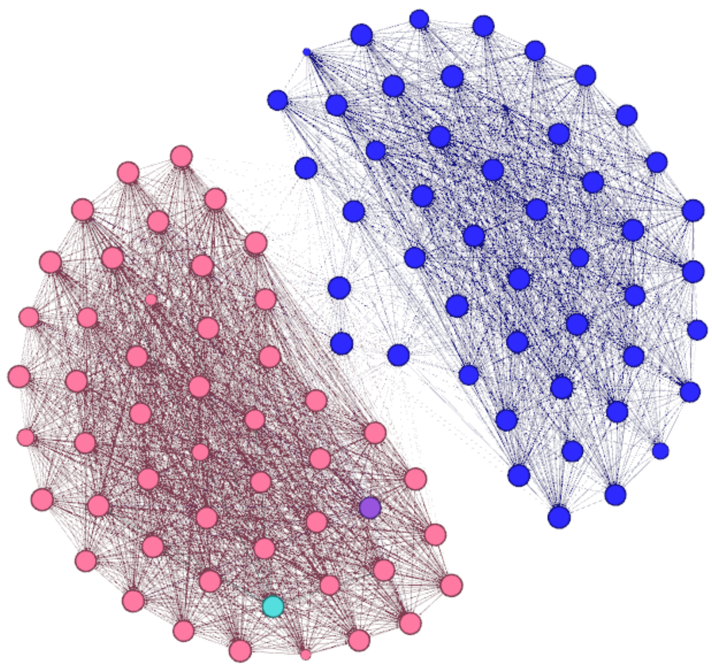

[Hansen 2011]

# Filtering

Results of Fruchterman-Reingold layout on
the 2007 US Senate voting data with and without edge filtering.



Without filtering



Edges corresponding to
% agreement < 0.65 are filtered

Node color represents party affiliation of a senator

# Understanding a Network

- Visualization alone cannot provide full understanding of a graph or network
- **Network graph metrics** are quantitative measures for describing a network, characterizing subgroups or specific nodes within a network
  - Influential people in a social network (e.g., celebrities in Twitter or Weibo)
  - Gatekeepers connecting communities (e.g., for headhunting in LinkedIn)

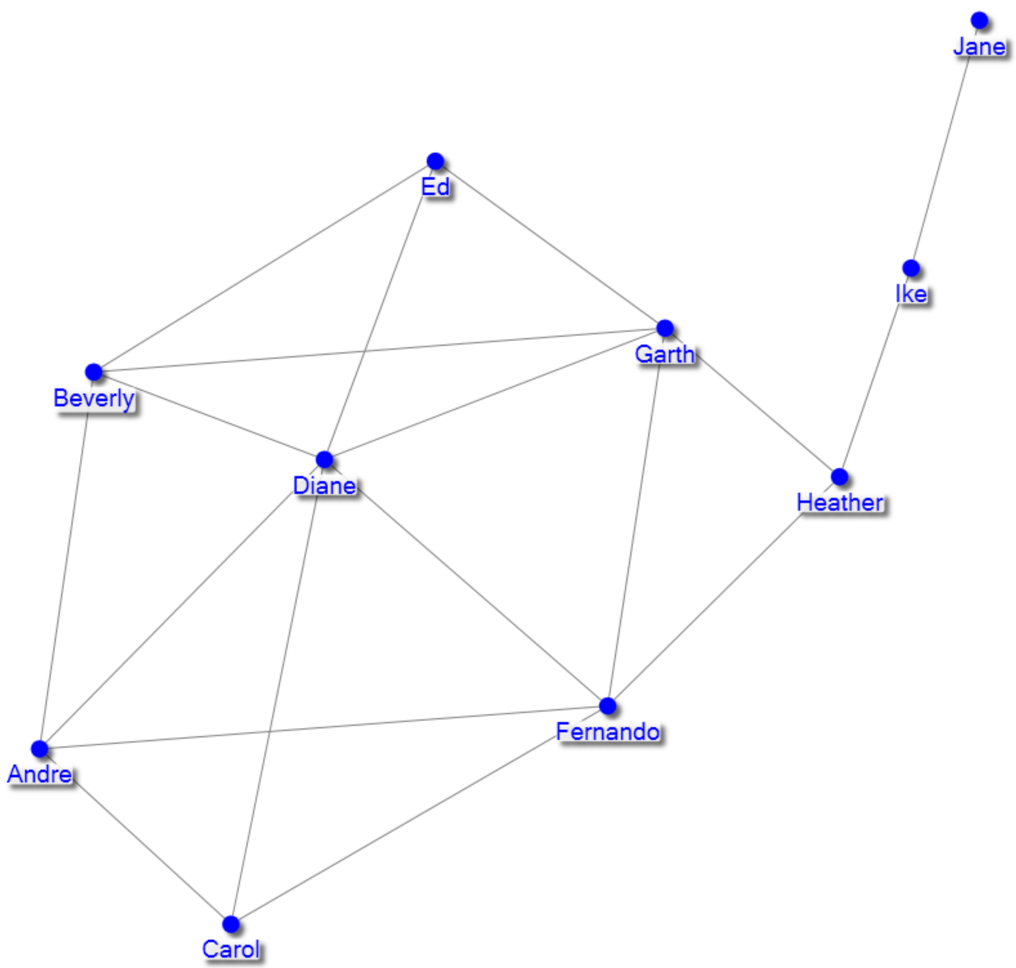
# Understanding a Network

- Studying network metrics over time helps understand how a network evolves
- Network metrics in turn improves the visual display of a network
  - By filtering and showing only the important nodes
  - By assigning different visual attributes to the nodes

# Network Metrics

- **Overall graph metrics**
  - Graph type / # of vertices / # of edges
  - Self loops   (e.g., a person replying his own emails)
  - Connected components
  - Isolated vertices
  - Maximum geodesic distance (aka diameter)
    - i.e., the distance between two nodes that are farthest apart
  - Average geodesic distance
    - i.e., average distance from one node to another through the graph edges
  - Graph density
    - i.e., # edges / max possible edges
  - etc.

# Overall Graph Metrics




- Max. geodesic dist. = 4
- Avg. geodesic dist. = 1.78
- Graph density = 18 / 45 = 0.4

# Network Metrics

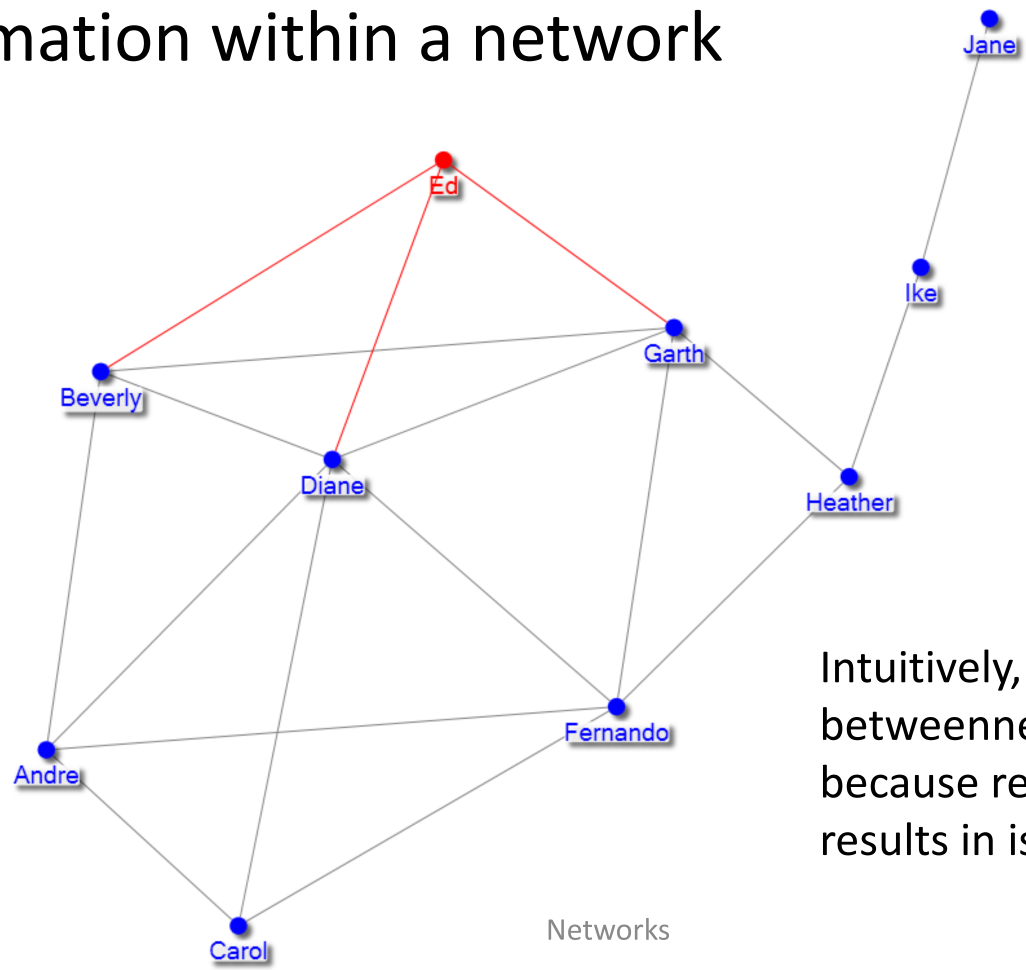
- **Node metrics**: structure-based metrics associated with a node
  - Degree (in-degree, out-degree)
  - Betweenness centrality
  - Closeness centrality
  - Eigenvector centrality
  - Clustering coefficient
  - PageRank
- Useful for **identifying special or important nodes** or subgroups
- There are **edge metrics** as well (e.g., edge betweenness)

# Degree Centrality

- **Centrality** means "Importance"
- Degree = number of neighbours
- For directed graphs
  - In-degree = number of incoming edges
  - Out-degree = number of outgoing edges

# Betweenness Centrality

- The importance of a person in passing information within a network




Intuitively, Heather is of higher betweenness centrality than Ed because removing Heather results in isolated nodes

# Betweenness Centrality

- The **betweenness centrality** of a node:

$$C(v) = \sum_{s,t \neq v \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

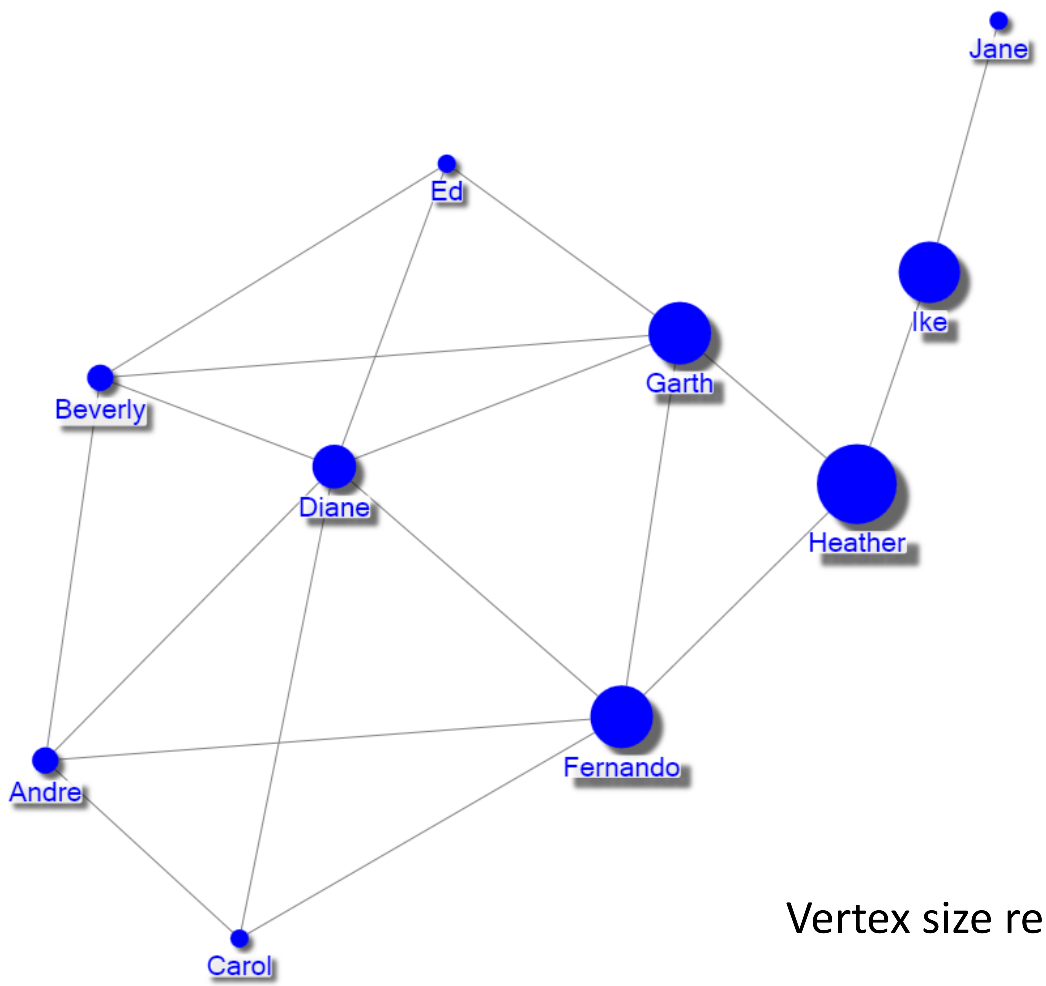
# of shortest path between *s* and *t* passing through *v*

Total # of shortest path between *s* and *t*

- Equivalent to computing the all-pairs shortest path of a graph — Complexity: $O(|V|^3)$

O(|V||E|) on unweighted sparse graph
Ulrik Brandes, "A Faster Algorithm for Betweenness Centrality ", Journal of Mathematical Sociology, 2001.

# Betweenness Centrality




Vertex size representing betweenness

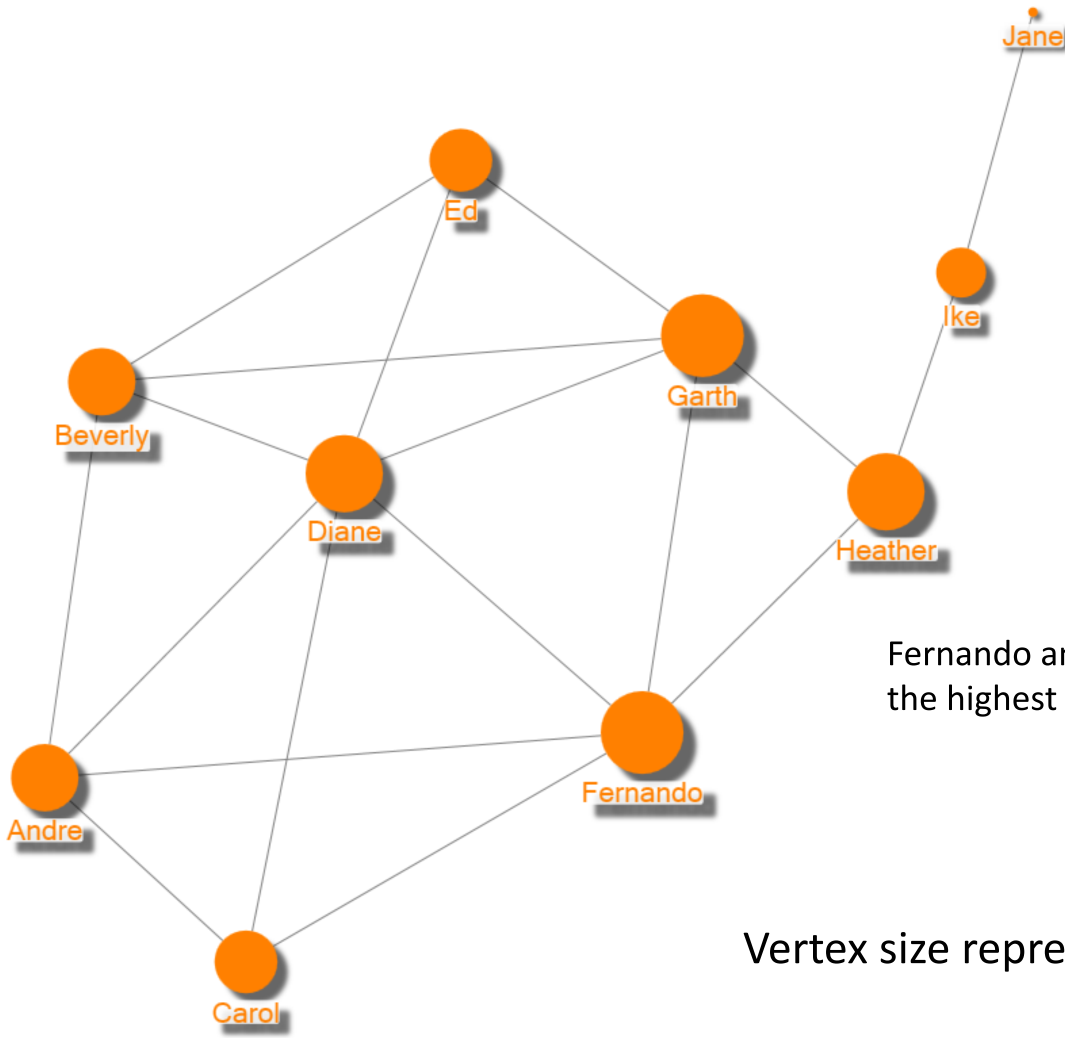
# Closeness Centrality

- Measures how close a person is to the others
  - how fast a message can reach all others from a person
- What is the fastest scenario for a person's message to reach all others?
- The **closeness centrality** of a node:

$$C(v) = \frac{1}{\sum_{u \neq v \in V} d(u, v)}$$

**Farness of v:**
Total distance between *v* and all other nodes

# Closeness Centrality

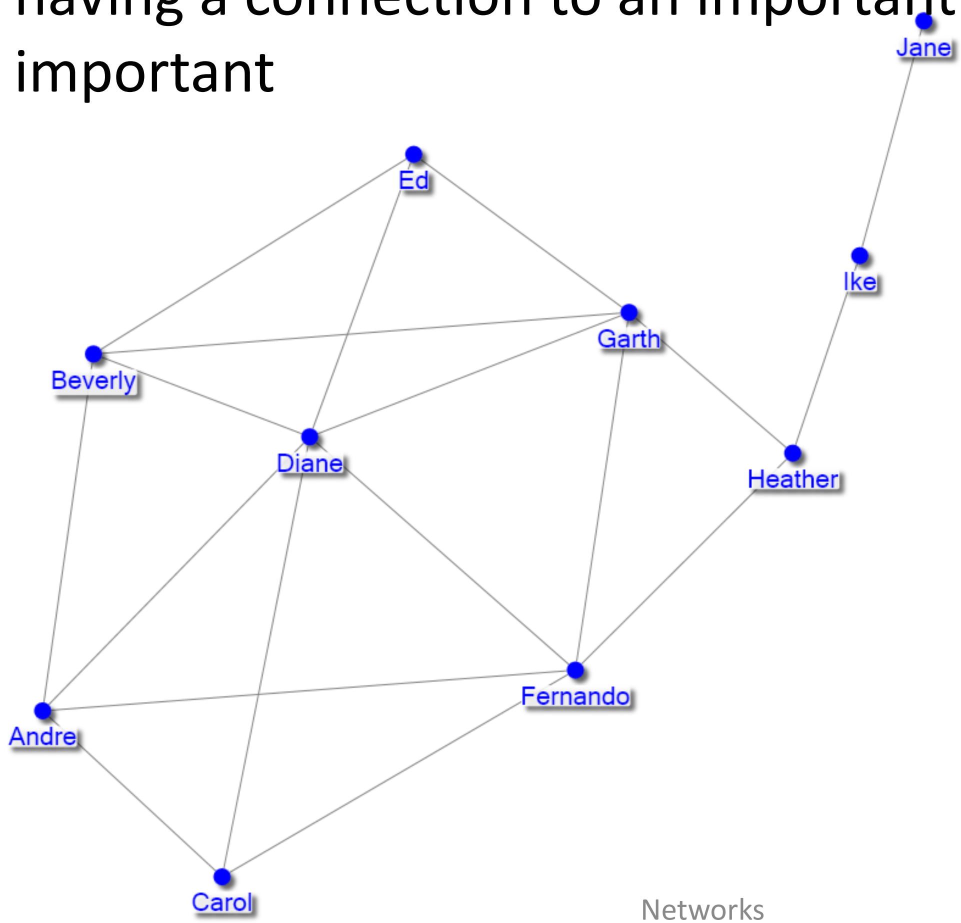



Fernando and Garth are of
the highest closeness measures

Vertex size representing closeness

# Eigenvector Centrality

- A measure of the influence of a node in a network
  - having a connection to an important person is more important




- deg(Heather) = deg(Ed) = 3
- Ed connects with Diane who is most popular (i.e., having the largest degree)
- Heather connects to Ike, who is among the least popular
- Hence, Ed's eigenvector centrality is higher

# Eigenvector Centrality

- The eigenvector centrality score of a node is:

$$x(u) = \frac{1}{\lambda} \sum_{v \in \mathcal{N}(u)} x(v) = \frac{1}{\lambda} \sum_{v \in V} a_{u,v} x(v)$$

Sum of scores of all its neighbours

$\lambda$ : constant

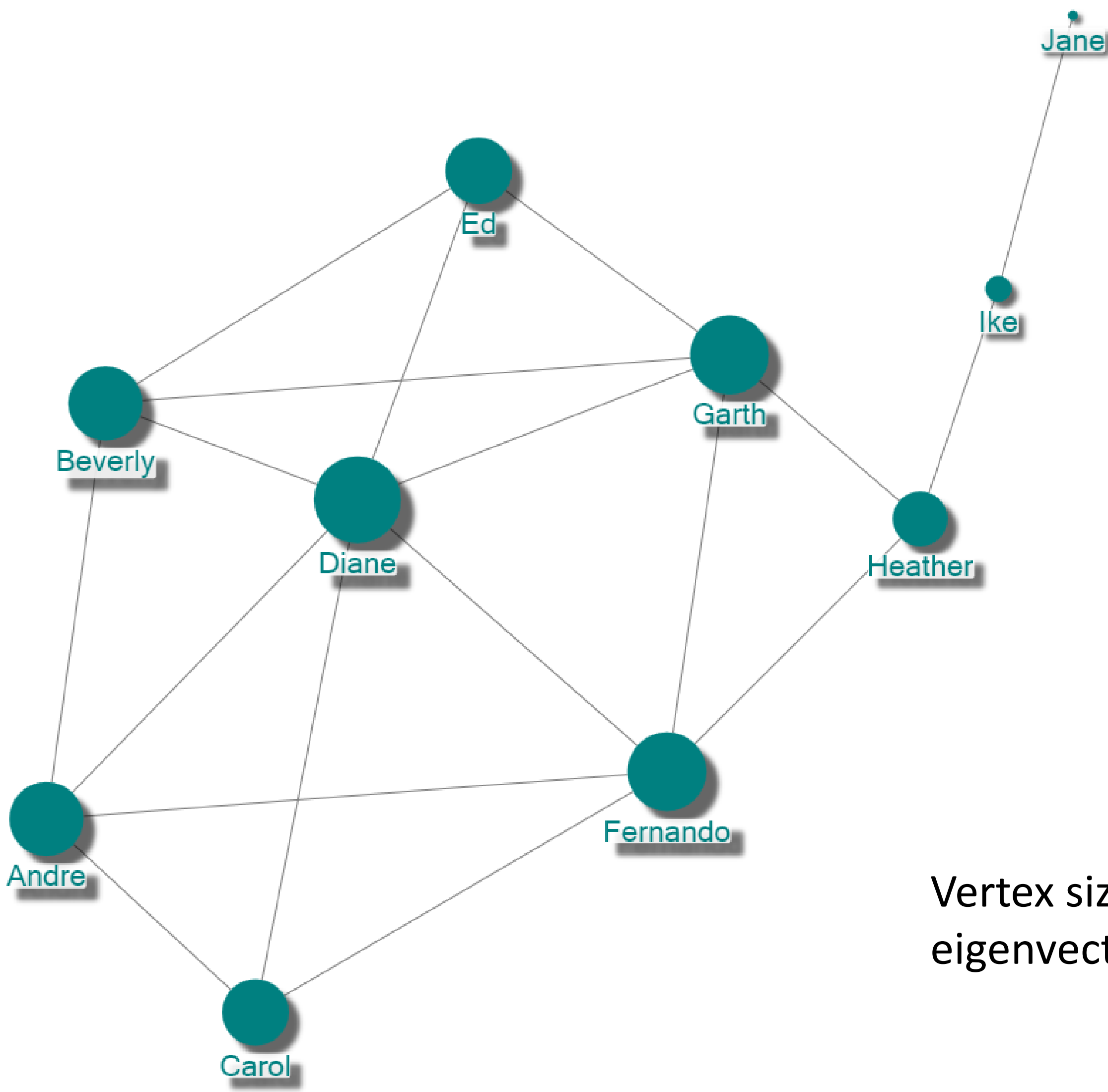
$\mathbf{A} = (a_{u,v})$: adjacency matrix

- Written in matrix form gives:

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

The eigenvector corresponding to the largest eigenvalue gives the scores

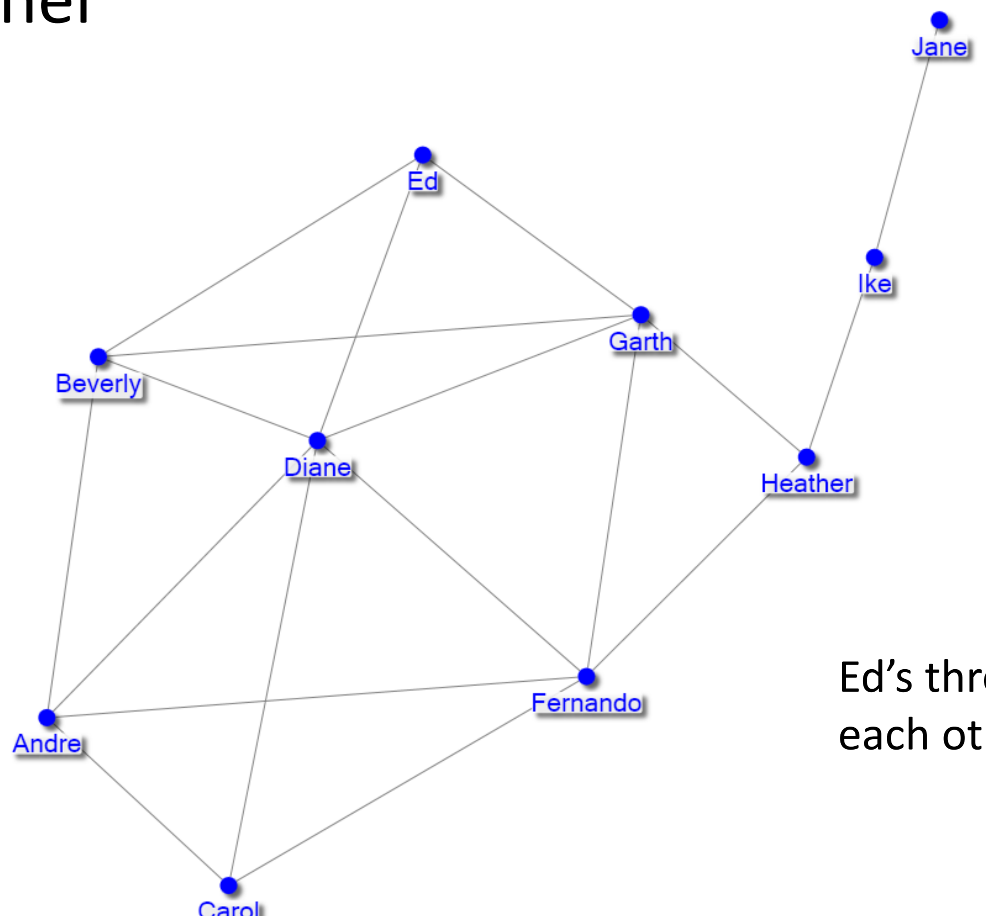
# Eigenvector Centrality




Vertex size representing eigenvector centrality

# Clustering Coefficient

- Measures how well a person's friends are connected to each other




Ed's three friends know each other

# Clustering Coefficient

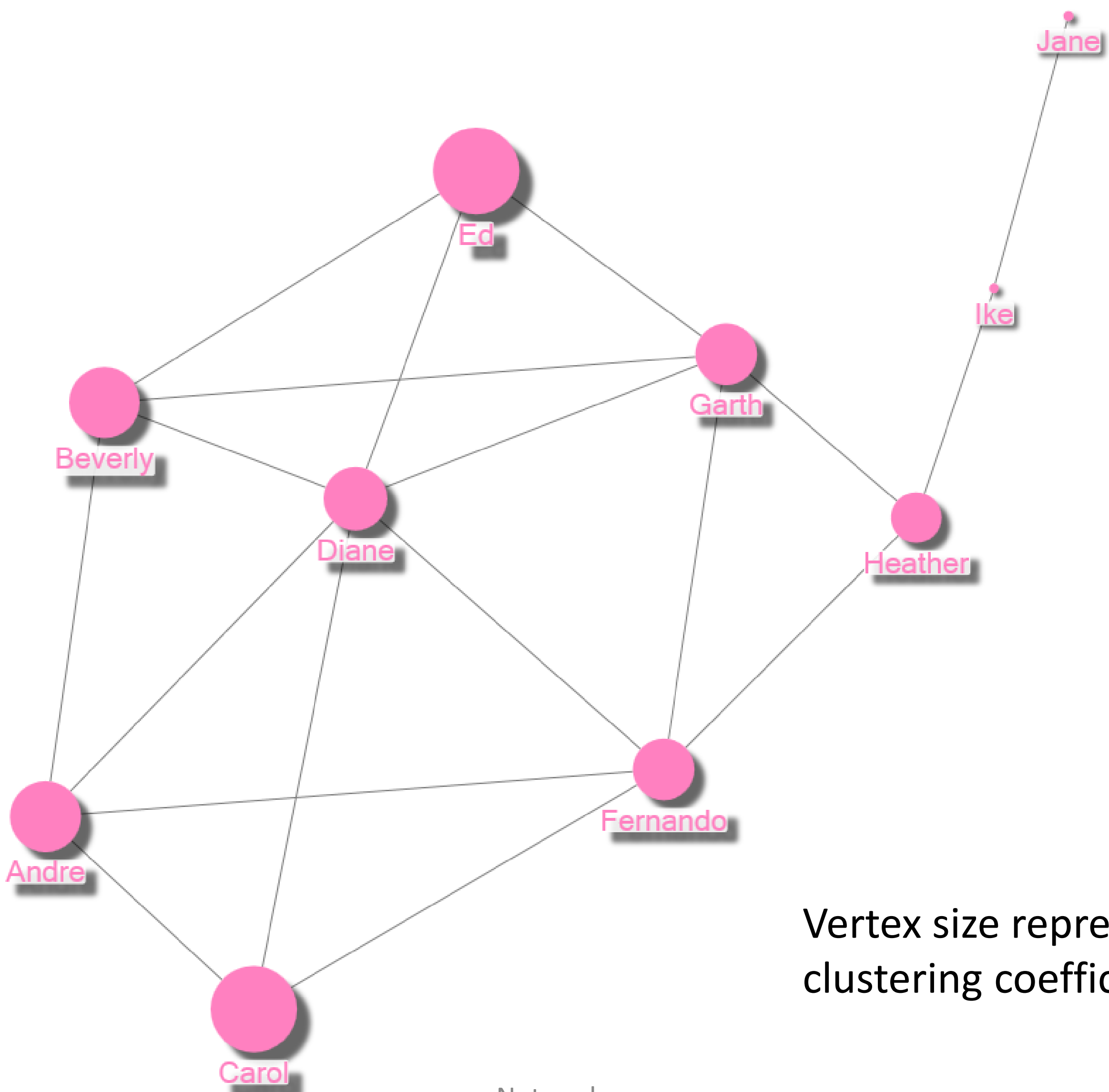
- Clustering coefficient of a node:

$$C(u) = \frac{\text{\# edges connecting } u\text{'s neighbours}}{\text{total \# possible edges connecting } u\text{'s neighbours}}$$

If $u$ has $k$ neighbours, this equals # of edges in a $k$-vertex clique (complete graph)
$= k \ (k-1) / 2$ for an undirected graph
$= k \ (k-1)$ for a directed graph

- Ranged in [0,1]
  - $C(u) = 0$: no edges among neighbours
  - $C(u) = 1$: neighbours form a clique

# Clustering Coefficient

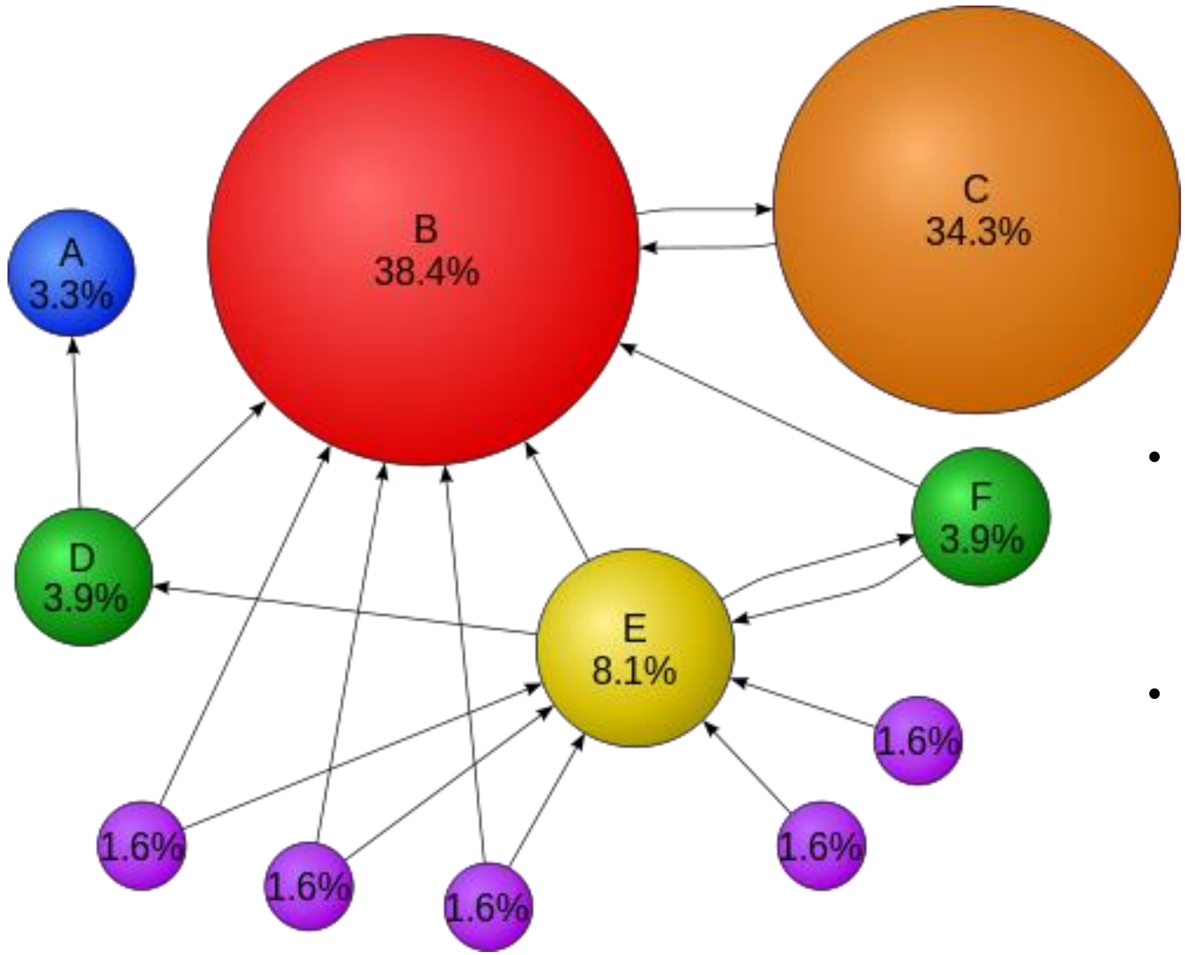



Vertex size representing clustering coefficient

# PageRank

- Used by Google search engine to rank websites in their search results.
- Idea:  More important websites are likely to receive more links from other websites.
- A variant of eigenvector centrality
- **Score of a page** = the probability of being brought to a page after many clicks.

# PageRank




http://en.wikipedia.org/wiki/PageRank

- The percentage shows the likelihood that a page can be reached after many clicks.
- Assumption: A user start on a random page, and has 85% chance of clicking randomly on any link in the page, and 15% chance of jumping randomly to any other page in the WWW.

# PageRank

- An assumption that there is possibility a surfer will stop following links and jump instead to a random page

$$\mathbf{M}\mathbf{x} = \mathbf{x}$$

where $$\mathbf{M} = p\mathbf{A} + \frac{(1-p)}{n}\,\mathbf{1}\,\mathbf{1}^T$$

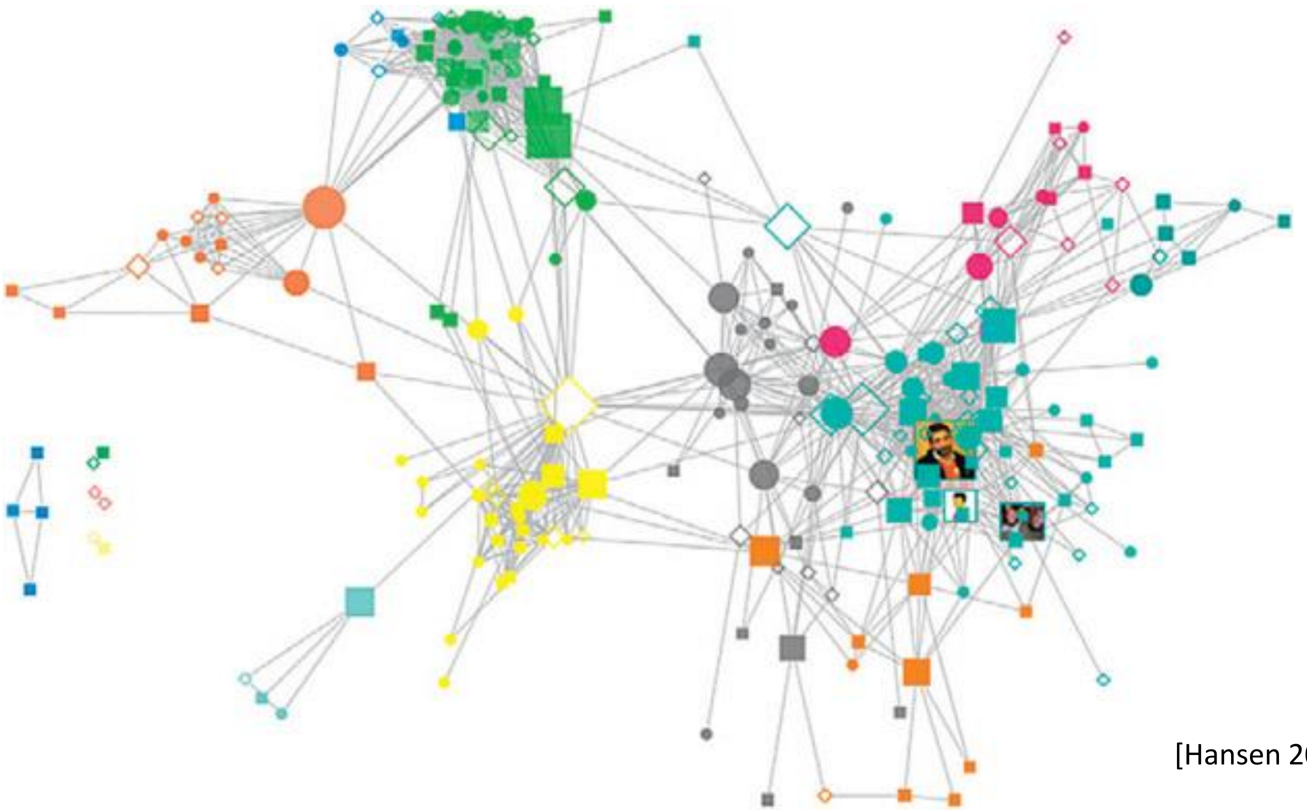
$1-p$ : probability of jumping to a random page

- Can be used for ranking nodes in a general network.

# Visual Attributes Mapping

- Map ordered visual attributes to ordered data and unordered visual attributes to categorical data
- Ordered data: Node degree, centrality
  Ordered visual attributes: Size, line width, opacity
- Categorical data: Gender, affiliation
  Unordered visual attributes: Color, shape

# Visual Attributes Mapping



[Hansen 2011]

shape ↔ gender
color ↔ cluster

size ↔ betweenness
opacity ↔ eigenvector centrality

# Tools and Datasets

- Tools for Graph Visualization
  - Gephi
    https://gephi.org
  - NodeXL
    http://nodexl.codeplex.com
- Large Network Datasets
  - A Citation Network Dataset
    http://arnetminer.org/citation
  - Stanford Network Analysis Project
    http://snap.stanford.edu

# Reference


- Ivan Herman, Guy Melançon, M. Scott Marshall, “Graph Visualization and Navigation in Information Visualization: A Survey”, *IEEE Trans. Vis. Comput. Graph*, 6 (1), 2000, pp. 24-43.
- Matthew Ward, Georges Grinstein and Daniel Keim, "*Interactive Data Visualization: Foundations, Techniques, and Applications*", 2010 [Chapter 8]
- Hansen, Shneiderman and Smith, “Analyzing Social Media Networks with NodeXL: Insights from a Connected World”, 2011.
- Isabel F. Cruz and Roberto Tamassia, “Graph Drawing Tutorial” (http://cs.brown.edu/~rt/papers/gd-tutorial/gd-constraints.pdf)