Network visualization methods

Outline

- Structural representation
  - Spring embedders
  - Other layouts

- Relational matrices representation
- Visualization syntax and attributes representation, network measures

Graph drawing

Graph drawing is not a trivial problem because they are not defined in a metric space

This has led to the development of many methods in literature.

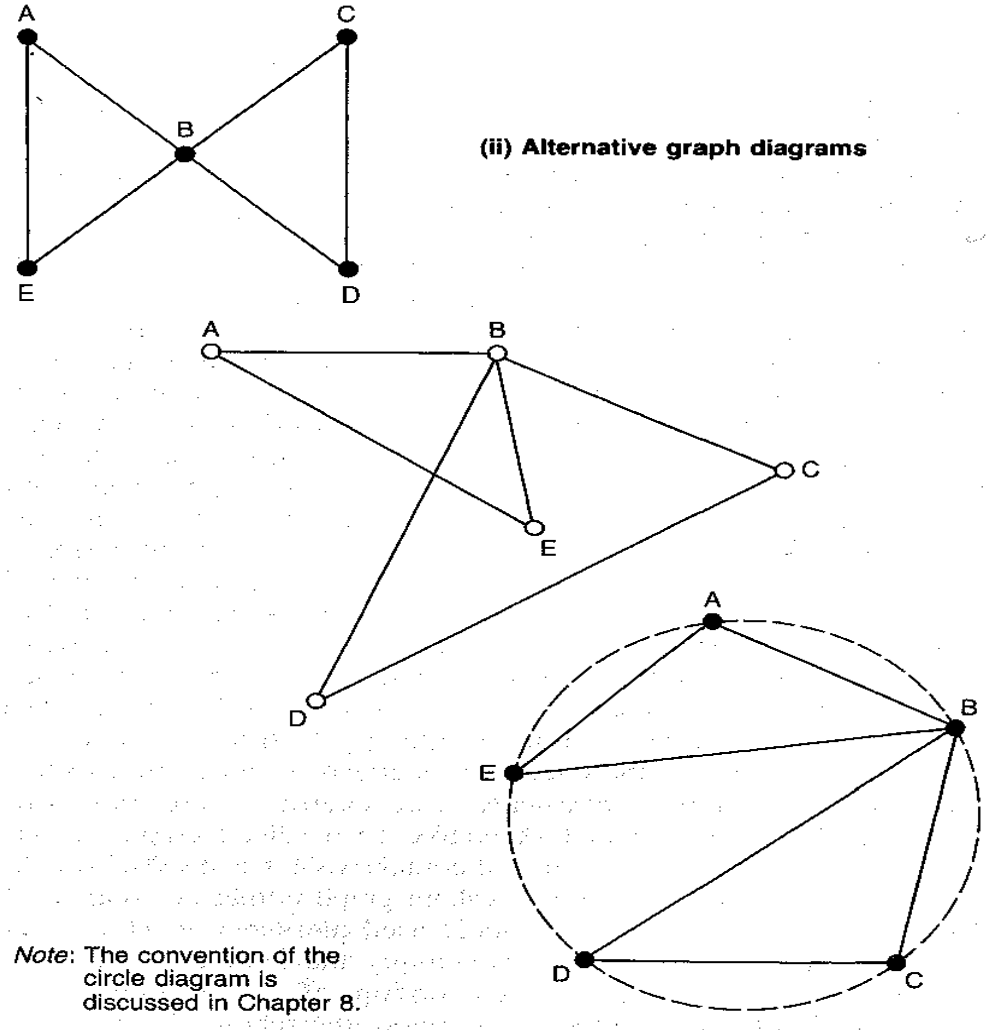
- some for general purpose
- others for specific purpose
- coming from different scientific community

(Social Network Community, Mathematical Community,

Statistical Community, Computer Scientists Community)

(i) Adjacency matrix

	A	B	C	D	E	Row sum
A	–	1	0	0	1	2
B	1	–	1	1	1	4
C	0	1	–	1	0	2
D	0	1	1	–	0	2
E	1	1	0	0	–	2
Column sum	2	4	2	2	2	



Note: The convention of the circle diagram is discussed in Chapter 8.

Figure 4.1 *Alternative drawings of a graph*

Graph visualization

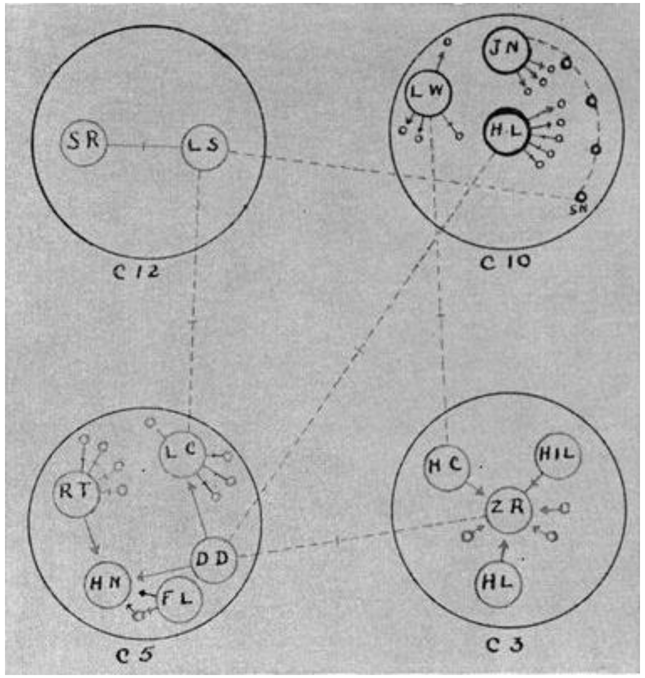
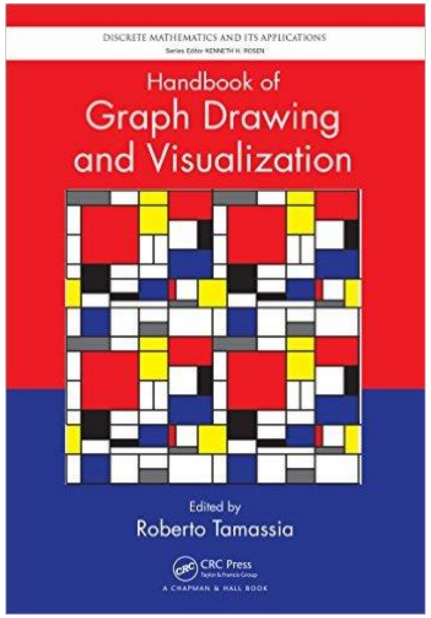


Figure 26.1 A sociogram from [Mor53, p. 422] showing a graph with fourteen highlighted vertices and four clusters.

Graph Visualization (Tomassia, 2013)



Network representations have two main goals:

- Data structure exploration
- Results communication

A good representation has to represent all relevant information from the analytical point of view.

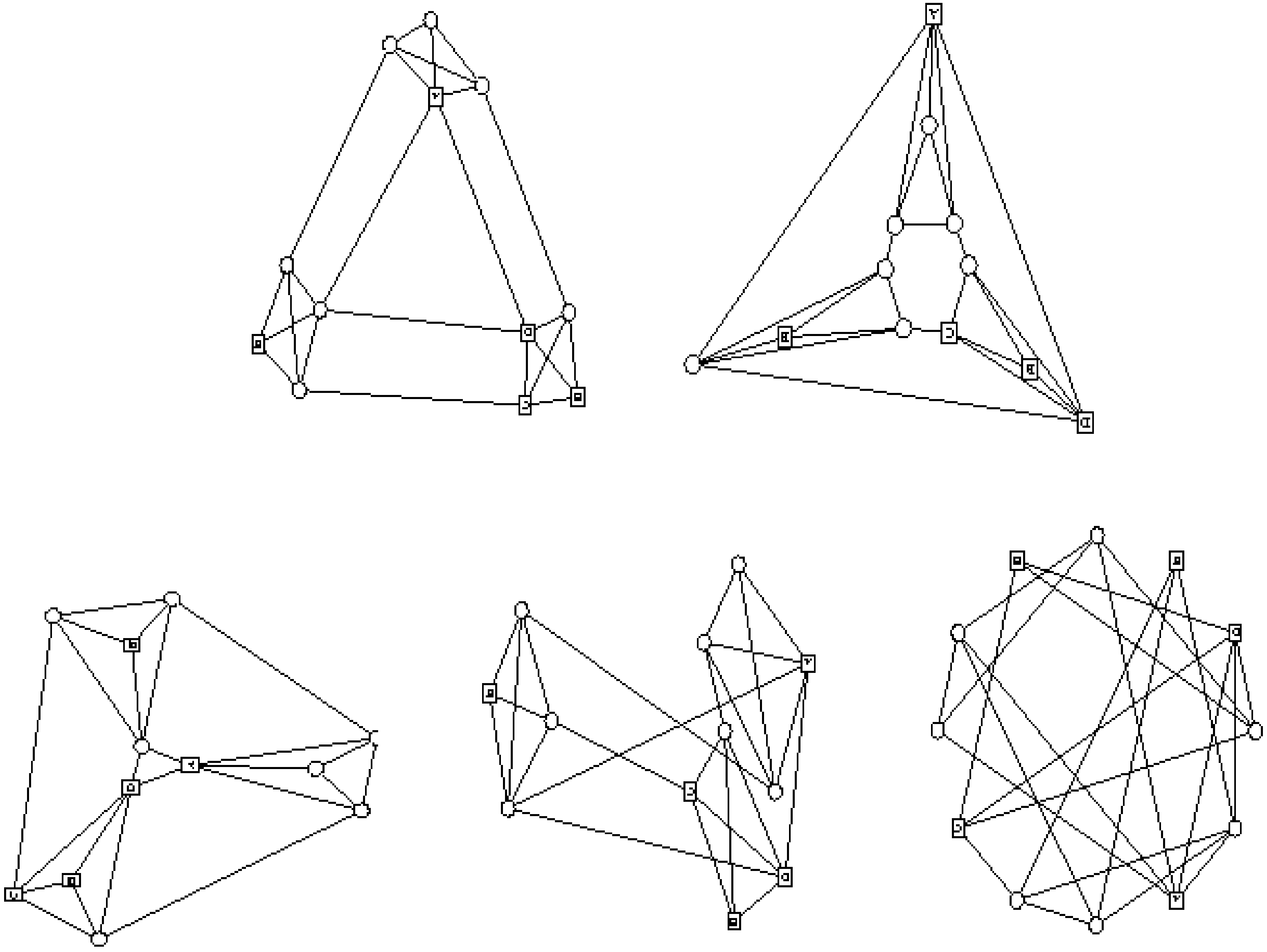
> "A process of charting has been devised by the sociometrists, the sociogram, which is more than merely a method of presentation. It is first of all a method of exploration." [Mor53, p. 95f]

Layout

- A drawing, or better a ***layout***, is a mapping of the nodes and edges into the plane (or into R3 for 3-dimensional drawings).

  - Nodes (actors, entities) are represented as a circles, points or other forms
  - Edges/Arcs (ties, relations) are represented as a segments or arrows
  - Information on nodes and edges can be visualized using text labels at various positions, different colors, or other visual elements such as thickness of lines, size of nodes

Layouts and information



- 5 examples of different layout of a given social network
- Each layout emphasized different aspects of the network
- The first layout emphasize the presence of 3 dense subgroups
- The last suggests a more interconnected system

Layout: Aesthetics criteria

- Crossings minimization

  If too many edges cross each other, the human eye can not easily find out which nodes are connected.
  If a graph can be drawn without edge crossings (such graphs are called planar) is often better

- Folds / curves minimization

  human eye can much more easily follow an edge with none or only a few bends. This is

- Graph area minimization

  a picture looks much better if the nodes and edges fill the space with homogenous density.

- Angle maximization

  It is important that edges are as far apart as possible for visual clarity especially in low resolution pictures.

- Edges length minimization

  Length of the segments representing edges should be minimized

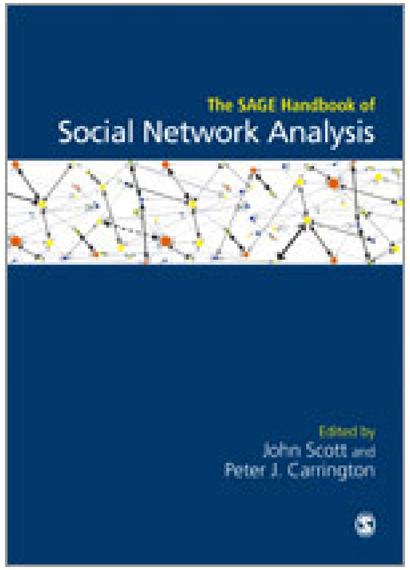
- Symmetry

  If graphs contain symmetrical substructures then it is important to show this symmetry in its layout.

- Clustering

  Especially in large networks is necessary to cluster the nodes to reveal some insights on the graph structures

Mapping networks (Krempel, 2009)

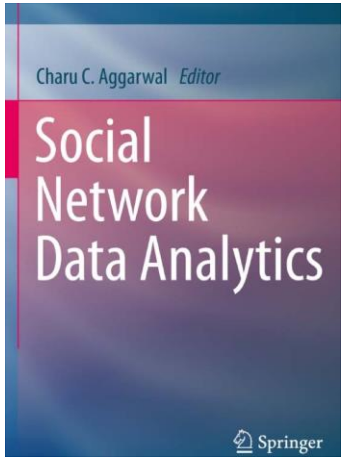


3 Mapping Networks

The most important task in mapping networks is to determine the 2D or 3D locations of the nodes from the links of a graph. Such a *layout* encodes certain features of a network that maintain as much information as possible relating to the embeddedness of the nodes.

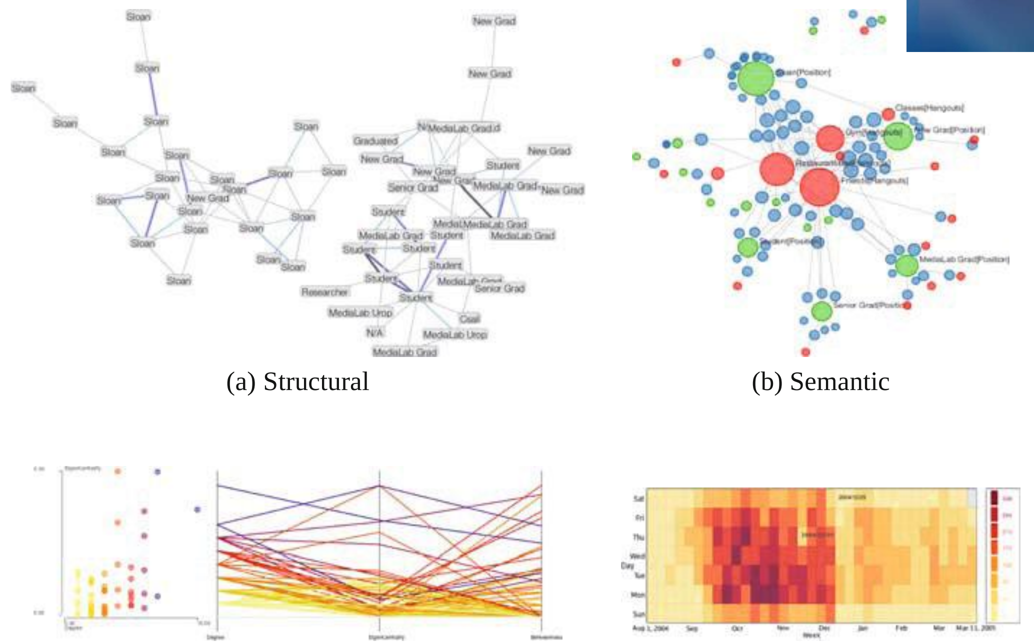
- Nodes position may reflects some network properties

A possible classification (Correa & Ma, 2011)



- Structural
  - Nodes-ties representation
    - Spring Embedders
    - Property Based Layout
    - Statistical Layout
  - (socio)Matrix-form
- Statistics
- Temporal or Dynamic methods

Visualizing Social Networks



(a) Structural (b) Semantic

(c) Statistical (d) Temporal

Figure 11.1. Example of different types of visualization (a) Structural, typically a node-link diagram. (b) Semantic, where nodes and links can represent different aspects of the social network. (c) Statistical, useful for depicting the distribution of social network metrics and (d) Temporal, a particular case of a semantic diagram that uses time as the main attribute.

C. C. Aggarwal (ed.), *Social Network Data Analytics,*
DOI 10.1007/978-1-4419-8462-3_1, © Springer Science+Business Media, LLC 2011

Nodes-ties structural representations

SPRING EMBEDDERS

structural representations of networks

A structural visualization of the social networks focuses precisely on that, its structure.

- The structure can be thought of as the **topology of a graph** represented only by nodes and edges in a social network.
- 2 predominant approaches to structural visualization:

1. node-link diagrams
2. matrix-oriented methods

structural representations of networks

Node-link diagrams are easy to interpret and depict explicitly the links between nodes (the most common representation of networks)

As social networks grow in complexity and size, finding a good layout becomes increasingly challenging

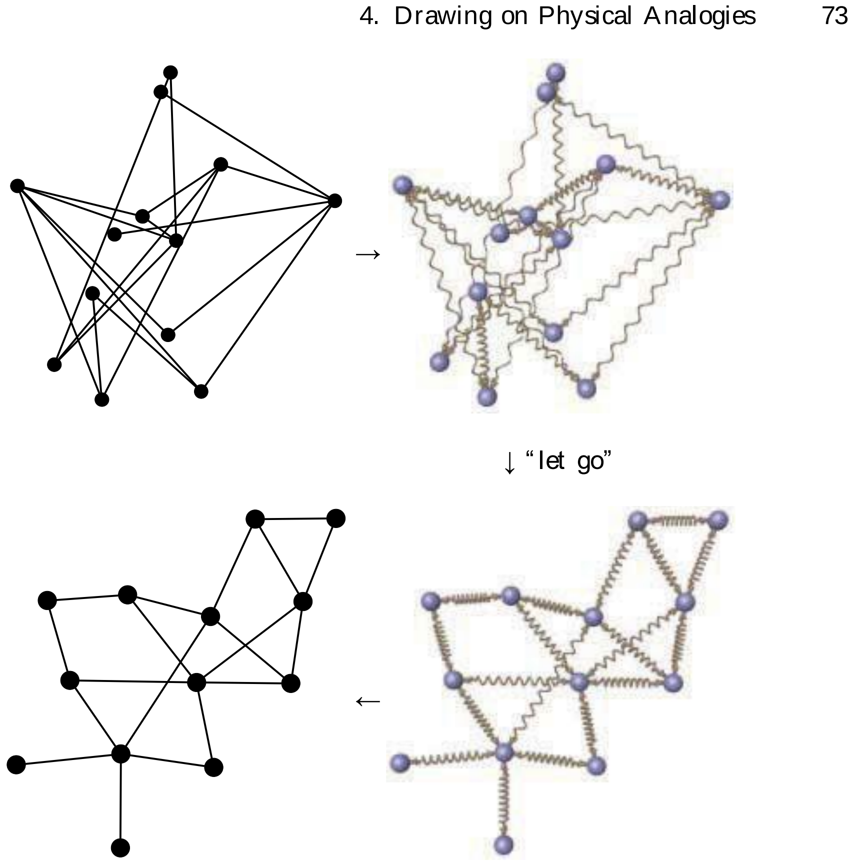
Some of the most flexible algorithms for calculating layouts of simple undirected graphs belong to a class known as force-directed algorithms, also known as spring embedders

Force-directed and Energy- Based Layout (spring embedders)

A class of methods applicable to general graphs, without prior knowledge of any structural properties, rather than relying on domain-specific knowledge.

Comparing the graph to a system of interacting physical objects, i.e. the springs and rods that link the spheres

The underlying assumption being that relaxed (energy-minimal) states of suitably defined systems correspond to readable layouts.



4. **Drawing on Physical Analogies**

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 200

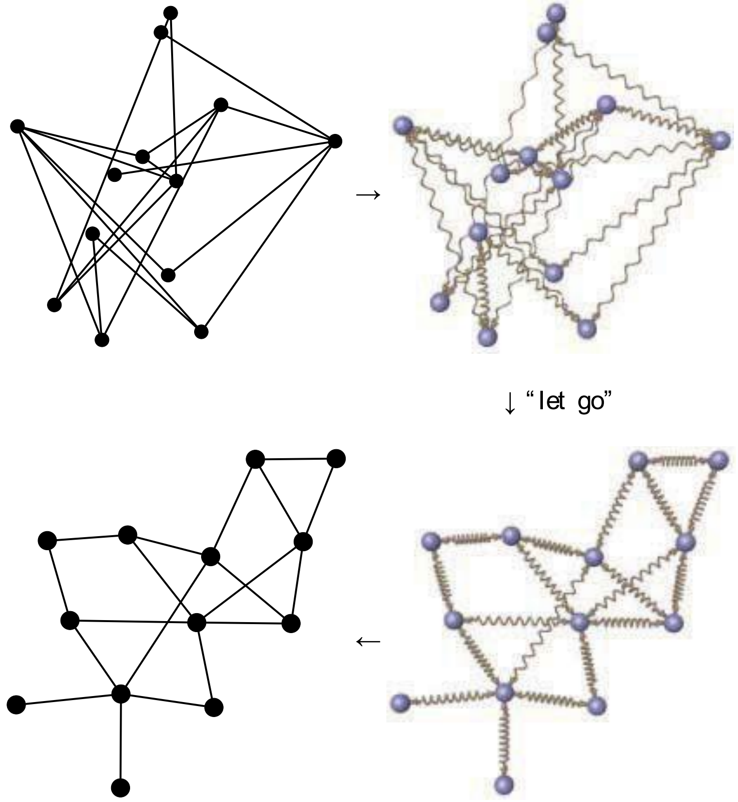
Force-directed and Energy- Based Layout

In general, these methods consist of two components:

1) a model consisting of physical objects (representing the elements of the graph) and interactions between these objects, and

2) an algorithm that (approximately) computes an equilibrium configuration of the system.

4. Drawing on Physical Analogies 73



4. Drawing on Physical Analogies

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 200

Force-directed and Energy- Based Layout

In these algorithms models employ the physical analogies → springs and rods that link spheres.

The attraction and repulsion forces that produce the final layout are designed to

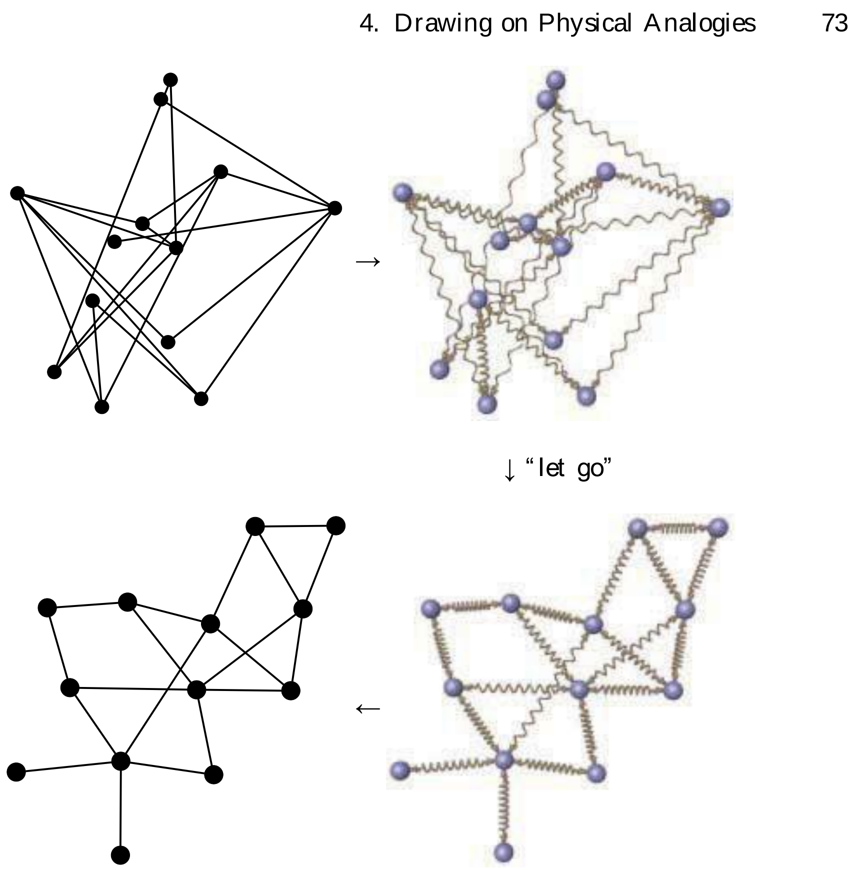
1.Distribute the vertices evenly in the frame.

2. Minimize edge crossings.

3. Make edge lengths uniform.

4. Reflect inherent symmetry.

5. Conform to the frame



4. Drawing on Physical Analogies

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders Force-directed (Eades 1984)

Given a graph *G(V,E)* and $p_v = (x_v, y_v)$ a vector of coordinates of a node v in the plane.

the attraction and repulsion forces can be defined as follows:

$$f_{\text{rep}}(p_u, p_v) = \frac{c_\varrho}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_u p_v} \qquad f_{\text{spring}}(p_u, p_v) = c_\sigma \cdot \log \frac{\|p_u - p_v\|}{l} \cdot \overrightarrow{p_v p_u},$$

with

$\|p_v - p_u\|$ the Euclidean distance between the position of two nodes *u* and *v*

$\overrightarrow{p_v p_u}$ the unit vector of the direction from p_v to p_u

c_ρ repulsion constant

c_σ spring force (attraction constant)

l the length of the springs

Spring Embedders Force-directed (Eades 1984)

The question is how to obtain an equilibrium configuration.

The Spring embedders algorithm works as follows:

- Node positions not corresponding to a system at equilibrium imply **positive internal stress**.
- The system reaches the equilibrium when the forces acting on each node are balanced.
- To relax a stressed system, vertices are iteratively moved, at time t, according to a **net force vector** $\mathrm{F_v}(t)$

  which is the sum of all repulsion and spring forces acting on a node *v*.

  After computing $\mathrm{F_v}(t)$for all $v \in V$, each node is moved a constant δ times this vector.
- By iteratively computing the forces on all nodes and updating positions accordingly, the

  system approaches a stable state, in which no local improvement is possible.

Spring Embedders Force-directed (Eades 1984)

Algorithm 6: Spring embedder

Input: connected undirected graph $G = (V, E)$
initial placement $p = (p_v)_{v \in V}$

Output: placement p with low internal stress

for $t \leftarrow 1$ *to* ITERATIONS **do**
  for $v \in V$ **do**
    $F_v(t) \leftarrow \sum_{u:\{u,v\} \notin E} f_{\text{rep}}(p_u, p_v) + \sum_{u:\{u,v\} \in E} f_{\text{spring}}(p_u, p_v)$
  for $v \in V$ **do** $p_v \leftarrow p_v + \delta \cdot F_v(t)$

Spring Embedders Force-Directed Fructherman & Reingold (1991)

To simplify calculations and to make the graph more readable, Fructherman and Reingold (1991) modified the repulsion and attraction forces computation as follows:

$$f_{\text{rep}}(p_u, p_v) = \frac{l^2}{\|p_u - p_v\|} \cdot \overrightarrow{p_u p_v} \qquad f_{\text{attr}}(p_u, p_v) = \frac{\|p_u - p_v\|^2}{l} \cdot \overrightarrow{p_v p_u}$$

The force of the spring between two adjacent vertices is given by the sum of the repulsion and attraction forces.

Spring Embedders Force-Directed Fructherman & Reingold (1991)

From the paper of Fruchterman and Reingold (1991):

“Eades modelled a graph as a physical system of rings and springs, but his implementation did not reflect Hooke’s law; rather, he chose his own formula for the forces exerted by the springs.”

Only two principles for graph drawing:

1. Vertices connected by an edge should be drawn near each other.

2. Vertices should not be drawn too close to each other.

How close vertices should be placed depends on how many there are and how much space is available

Spring Embedders Force-Directed Fructherman & Reingold (1991)

The attractive and repulsive forces are redefined as:

$$f_a(d) = d^2/k, \qquad f_r(d) = -k^2/d,$$

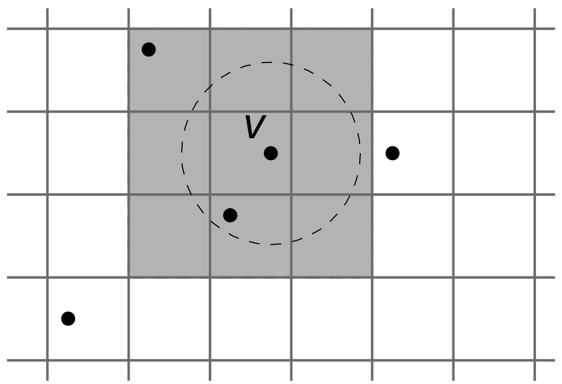
in terms of the distance d between two vertices and the optimal distance between vertices

k defined as

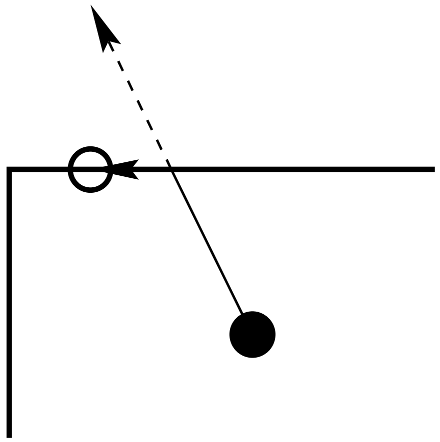
$$k = C\sqrt{\frac{area}{number\ \ of\ \ vertices}}$$

Spring Embedders Force-Directed Fructherman & Reingold (1991)

The Eades algorithm was also modified by introducing other correction factors:



(a) neglecting weak repulsive forces



(b) coordinate clipping

(a) Since repulsion from far away vertices does not contribute much to the displacement vector, a grid is established and the repulsion force is calculated only for nodes within the same grid cell.

(b) The displacement ensures that the graph is laid out inside a rectangular area, like a screen. If the displacement positions a vertex beyond a fixed boundary, the coordinate of the displacement vector is clipped.

4. Drawing on Physical Analogies

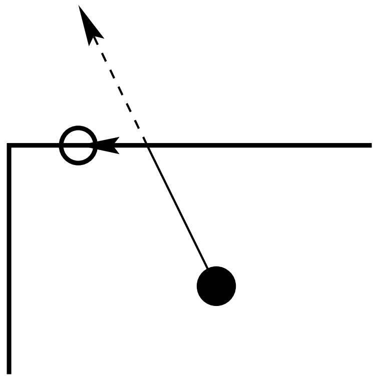
Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Fructherman & Reingold (1991) – GRID VARIANT

The parameter defining the maximum length of the shift δ is modify at each iteration $\delta(t)$ so that when the system is reaching the equilibrium the shifts are smaller.

The direction of movement is constrained with respect to a rectangular area (the screen or sheet of paper) and when the node move close the edge of the area, the moving vector proceeds along the border/margin.



(b) coordinate clipping

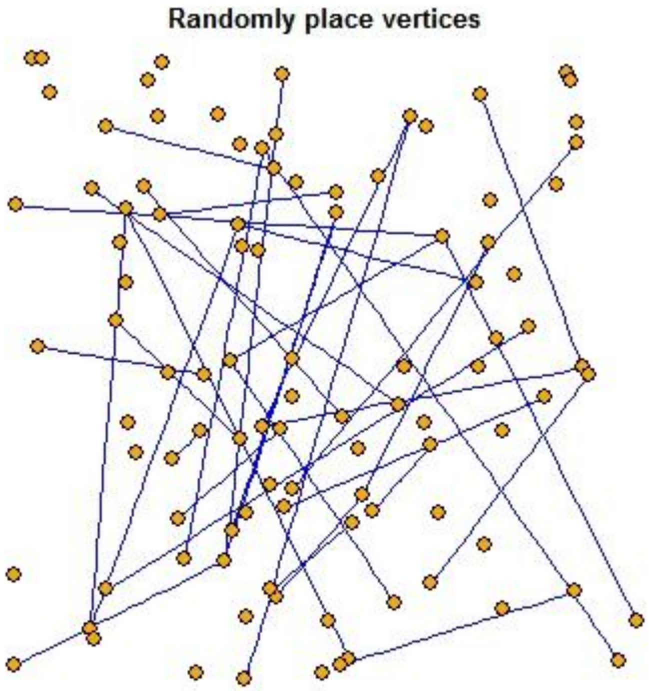
Fig. 4.4. Spring embedder modifications of Fruchterman and Reingold (1991).

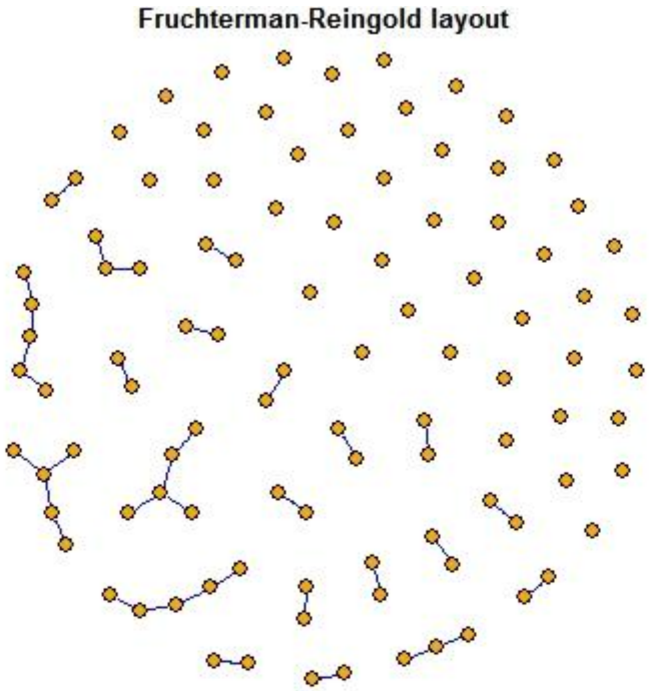
4. Drawing on Physical Analogies

Ulrik Brandes

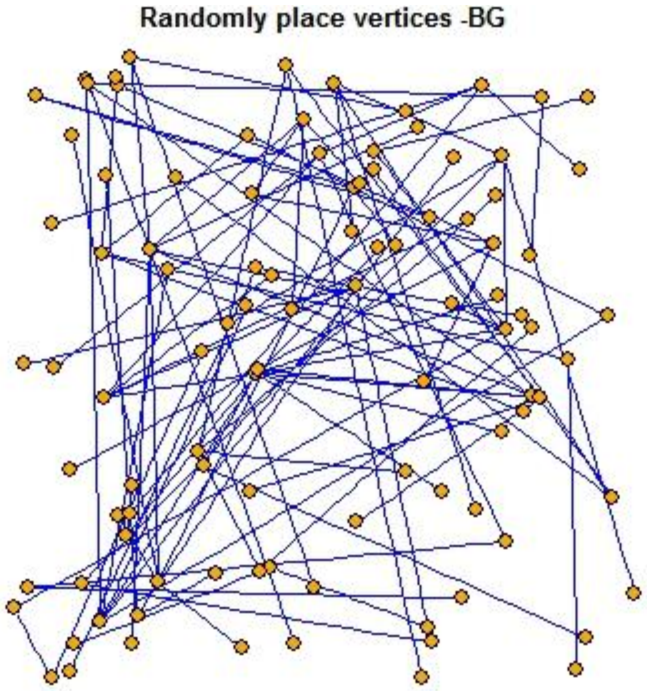
M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

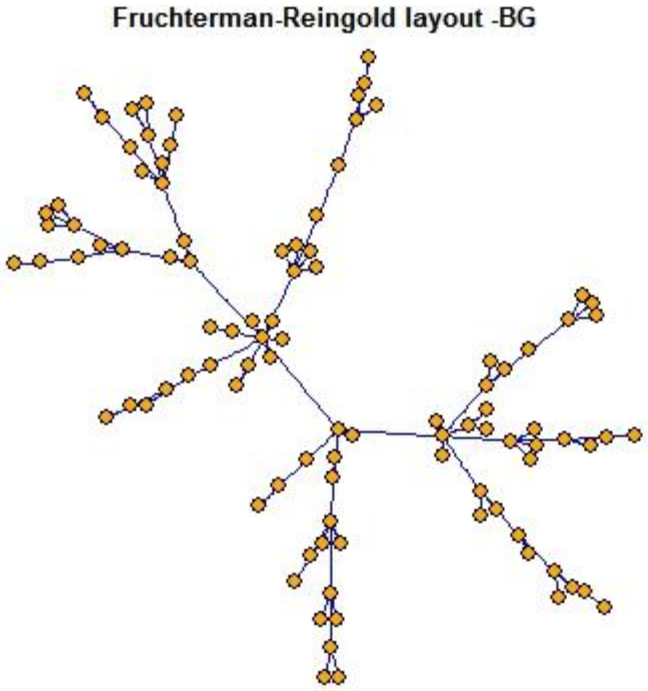
Spring Embedders Force-Directed Fructherman & Reingold (1991)





Spring Embedders Force-Directed Fructherman & Reingold (1991)





Spring Embedders Energy-based Kamada and Kawai (1989)

Force-directed algorithms that minimize the force acting on each node have the implicit goal of minimizing the overall energy of the system (given by the sum of the forces not in equilibrium).

Energy-based algorithms try to minimize the energy of the system directly.

Given a spring of force c_σ, natural length l and real length *d,* the spring will have a potential energy equal to:

$$U_{\text{spring}}(d) = c_\sigma \cdot (d - l)^2$$

4. **Drawing on Physical Analogies**

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders Energy-based Kamada and Kawai (1989)

K&K assume that:

- The spring ideal length between two nodes is given by the minimum path, geodesic distance $d_G(u,v)$, multiplied for the edge length
- The best paths in the graph are linear (as if they were Euclidean distances among the positions)
- Under these assumptions the objective function that results is as follows:

$$U_{KK}(p) = \sum_{u,v \in V} \frac{c}{d_G(p_u, p_v)^2} \cdot \left(\| p_u - p_v \| - l \cdot d_G(u,v) \right)^2$$

In this model there are no separate attractive and repulsive forces between pairs of vertices, but instead if a pair of vertices is (geometrically) closer/farther than their corresponding graph distance the vertices repel/attract each other.

4. Drawing on Physical Analogies

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders Energy-based Kamada and Kawai (1989)

The goal of the algorithm is to find values for the variables that minimize the objective function (energy function)

At local minimum all the partial derivatives are 0 and this corresponds to solving a number of simultaneous non linear equations (2*number of vertices)

KK computes the position of one vertex at time viewing U as function of only p_u and p_v

The objective function then can be minimized through a modified Newton-Raphson algorithm, where (from time to time) the node with the highest gradient moves.

4. Drawing on Physical Analogies

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders Energy-based Kamada and Kawai (1989)

The 1989 algorithm of Kamada and Kawai introduced a different way of thinking about "good" graph layouts.

-the algorithms of Eades and Fruchterman-Reingold aim to keep adjacent vertices close to each other while ensuring that vertices are not too close to each other, Kamada and Kawai take graph theoretic approach.

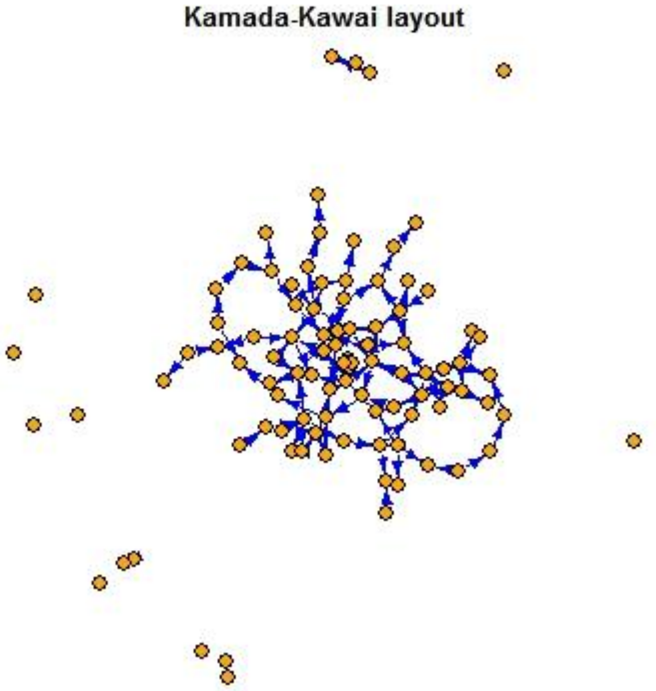
In this model, the "perfect" drawing of a graph would be one in which the pair-wise geometric distances between the drawn vertices match the graph theoretic pairwise distances and there are no separate attractive and repulsive forces between pairs of vertices, but instead if a pair of vertices is (geometrically) closer/farther than their corresponding graph distance the vertices repel/attract each other.

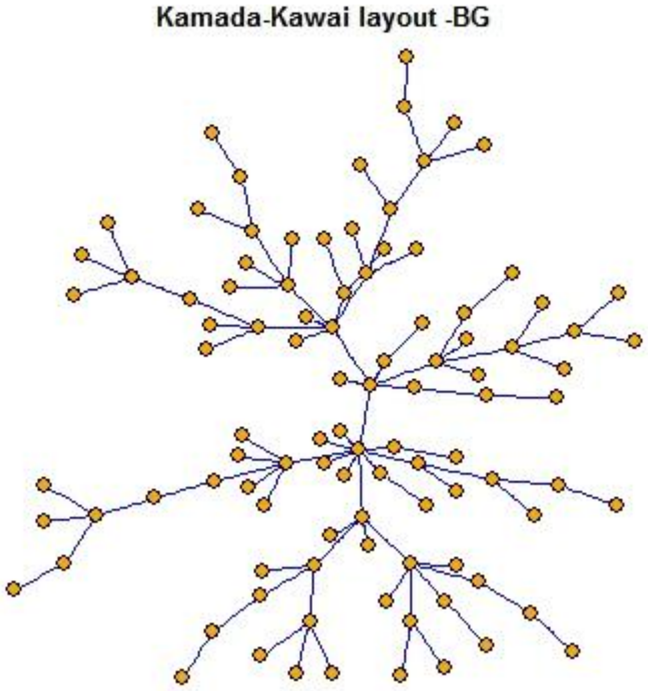
4. Drawing on Physical Analogies

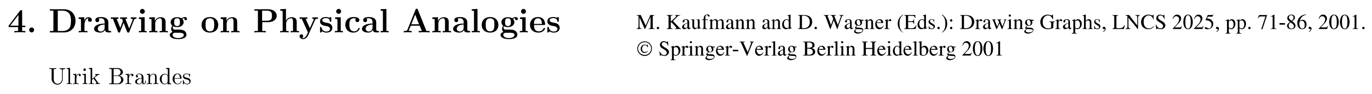
Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders Energy-based Kamada and Kawai (1989)







Spring Embedders for 2-mode networks

In the case of a 2-mode network is not possible to apply directly the spring embedding algorithms to the network but it is necessary data transformation.

If **I** is the incidence matrix actors for events is possible:

- To apply the spring embedding algorithms after the use of projection approach: Calculating the two weighed adjacency matrices *actors* x *actors* (**I** x **I**') or *events* x *events* ('**I** x **I**)
  - In this way we get two separate networks
- To apply the spring embedding algorithms to a block matrix constructed as follows $B = \begin{bmatrix} 0 & I' \\ I & 0 \end{bmatrix}$
  - In this way we get a single networks

4. Drawing on Physical Analogies

Ulrik Brandes

M. Kaufmann and D. Wagner (Eds.): Drawing Graphs, LNCS 2025, pp. 71-86, 2001.

Spring Embedders for 2-mode networks

Deep South Women (Davis,Gadner and Gadner, 1941)

Names of Participants of Group I	Code Numbers and Dates of Social Events Reported in *Old City Herald*													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson	×	×	×	×	×	×		×	×					
2. Miss Laura Mandeville	×	×	×		×	×	×	×						
3. Miss Theresa Anderson		×	×	×	×	×	×	×	×					
4. Miss Brenda Rogers	×		×	×	×	×	×	×						
5. Miss Charlotte McDowd			×	×	×		×							
6. Miss Frances Anderson			×		×	×		×						
7. Miss Eleanor Nye					×	×	×	×						
8. Miss Pearl Oglethorpe						×		×	×					
9. Miss Ruth DeSand					×		×	×	×					
10. Miss Verne Sanderson							×	×	×			×		
11. Miss Myra Liddell								×	×	×		×		
12. Miss Katherine Rogers								×	×	×		×	×	×
13. Mrs. Sylvia Avondale							×	×	×	×		×	×	×
14. Mrs. Nora Fayette						×	×		×	×	×	×	×	×
15. Mrs. Helen Lloyd							×	×		×	×	×		
16. Mrs. Dorothy Murchison								×	×					
17. Mrs. Olivia Carleton									×		×			
18. Mrs. Flora Price									×		×			

Spring Embedders for 2-mode networks (Borgatti)

Weighed adjacent matrices *actors* x *actors* (**I** x **I**’)

	EVE	LAU	THE	BRE	CHA	FRA	ELE	PEA	RUT	VER	MYR	KAT	SYL	NOR	HEL	DOR	OLI	FLO
EVELYN	8	6	7	6	3	4	3	3	3	2	2	2	2	2	1	2	1	1
LAURA	6	7	6	6	3	4	4	2	3	2	1	1	2	2	2	1	0	0
THERESA	7	6	8	6	4	4	4	3	4	3	2	2	3	3	2	2	1	1
BRENDA	6	6	6	7	4	4	4	2	3	2	1	1	2	2	2	1	0	0
CHARLOTTE	3	3	4	4	4	2	2	0	2	1	0	0	1	1	1	0	0	0
FRANCES	4	4	4	4	2	4	3	2	2	1	1	1	1	1	1	1	0	0
ELEANOR	3	4	4	4	2	3	4	2	3	2	1	1	2	2	2	1	0	0
PEARL	3	2	3	2	0	2	2	3	2	2	2	2	2	2	1	2	1	1
RUTH	3	3	4	3	2	2	3	2	4	3	2	2	3	2	2	2	1	1
VERNE	2	2	3	2	1	1	2	2	3	4	3	3	4	3	3	2	1	1
MYRNA	2	1	2	1	0	1	1	2	2	3	4	4	4	3	3	2	1	1
KATHERINE	2	1	2	1	0	1	1	2	2	3	4	6	6	5	3	2	1	1
SYLVIA	2	2	3	2	1	1	2	2	3	4	4	6	7	6	4	2	1	1
NORA	2	2	3	2	1	1	2	2	2	3	3	5	6	8	4	1	2	2
HELEN	1	2	2	2	1	1	2	1	2	3	3	3	4	4	5	1	1	1
DOROTHY	2	1	2	1	0	1	1	2	2	2	2	2	2	1	1	2	1	1
OLIVIA	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	2	2
FLORA	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	2	2

Figure 2. Women-by-women matrix of overlaps across events.

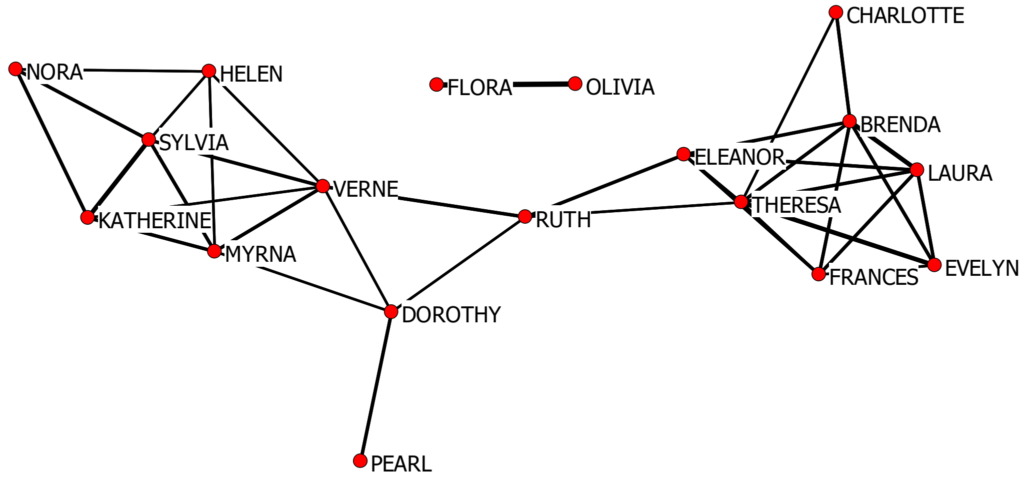


Figure 6. Spring-embedding representation of Jaccard similarities dichotomized at > 0.4

Spring Embedders for 2-mode networks (Borgatti)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8. Bipartite adjacency matrix B created from the original DGG 2-mode matrix X.

Block matrix $B = \begin{bmatrix} 0 & I' \\ I & 0 \end{bmatrix}$

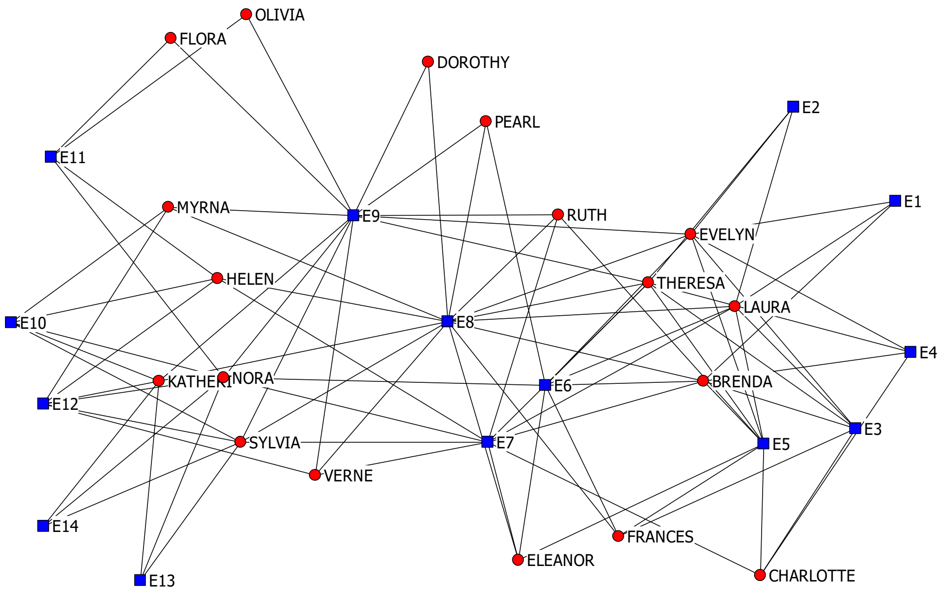


Figure 9. Spring-embedding representation of bipartite graph.

Advantages of Spring Embedders

- Produce very readable graphs for networks that are not excessively large
- Distribute the nodes uniformly and try to have similar edge length
- Visualization is often more effective in 3D and if supported by graphing interaction tools (manual correction)
- They are very useful if you want to proceed to a classification and grouping: spring embedders bring out the dense subgraphs very well
- If you add vertices that represent clusters, the representation becomes even more effective
- There are numerous variations in literature that represent ties with curved line and introduce constraints on representation and orientation

Disadvantages of Spring Embedders

They are not able to represent

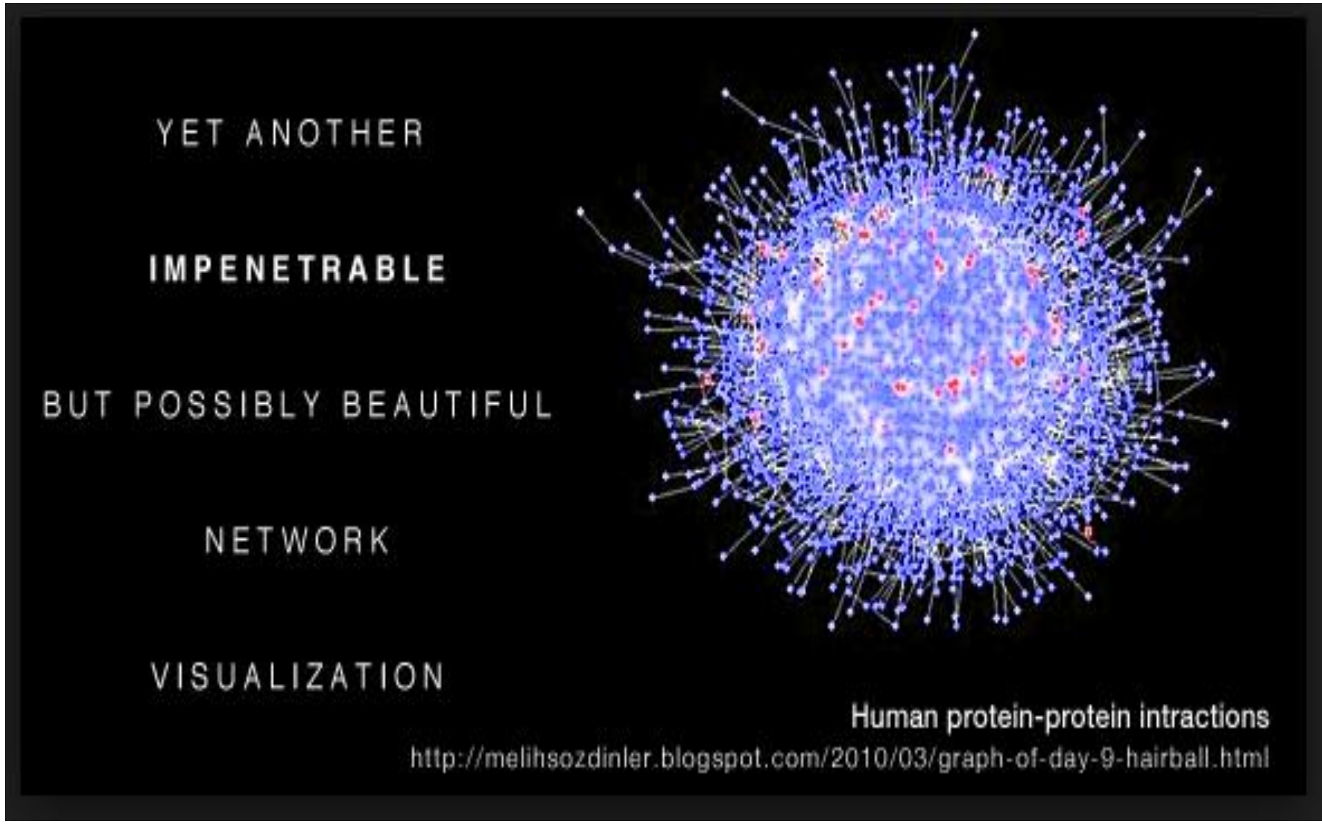
- Structural aspects
- Roles
- Positions
- Attributes

They can have very high computational costs

They depend on the starting position (random)

They consider symmetric matrices

Often they end up reproducing the 'hairball effect'



Nodes-ties structural representations

Multidimensional Scaling (MDS)

The MDS is a statistical method that starts from a matrix of distances or dissimilarities among n objects trying to reconstruct a coordinate system (not known) of n objects.

The MDS minimizes a objective function called STRESS that represents the difference among the observed distances D_i and the distance obtained with the coordinates of objects $\mathbf{x}$ and $\mathbf{x}_j$:

$$\text{Stress}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \;=\; \left(\sum_{i \neq j = 1..N} (D_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 \right)^{1/2}$$

The vector size $\mathbf{x}$ is usually equal to 2 or 3 to get planar or spatial representations.

MDS for social networks

Given a 1-mode graph $G(V, E)$, we construct the matrix of the geodesic distances D the layout with the MDS is obtained by solving the minimization problem:

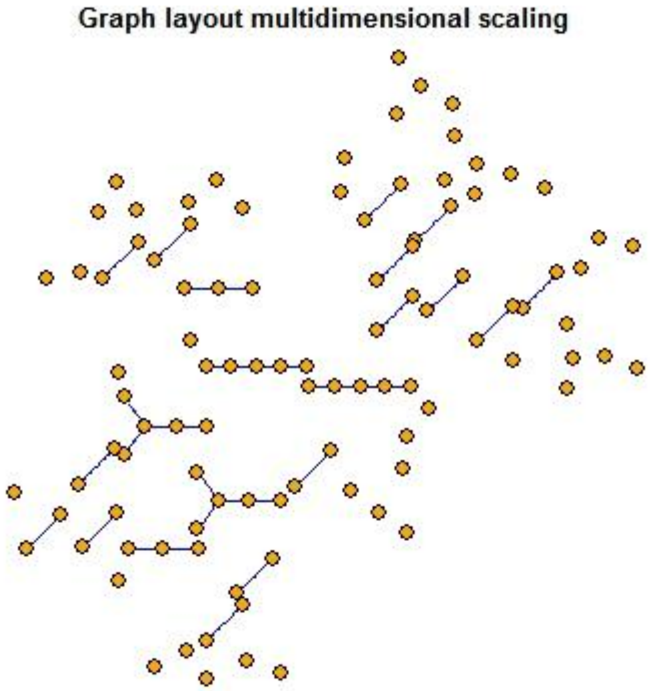
$$STRESS = \left(\sum_{u,v \in V} \left(d_G(u,v) - \|p_u - p_v\| \right)^2 \right)^{1/2}$$

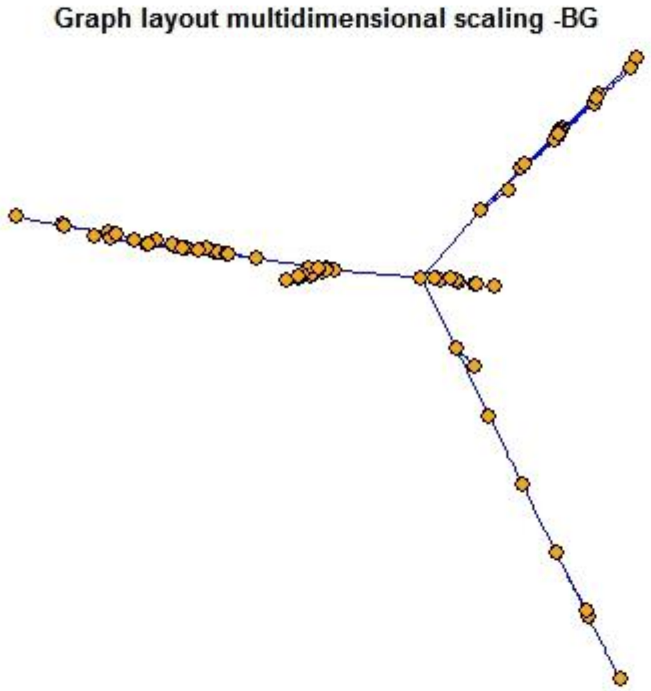
That can also be modified to be a spring embedder using a weighting factor is used to either emphasize or dampen the importance of certain pairs

It is interesting to note that this objective function resembles that of Kamada and Kawai where c=1.

Originally Kruskal minimizes the STRESS function using gradient descent but one can follow the KK approach (Newton-Raphson method)

MDS for social networks





MDS for social networks

Possible alternatives

- Apply the MDS algorithm to a modified adjacency matrix

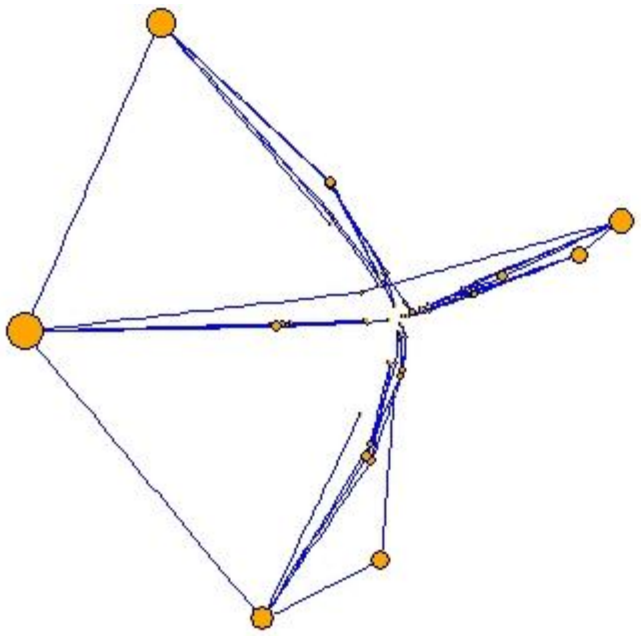
$$\boldsymbol{A}^* = \mathbf{1}\text{-}\boldsymbol{A}$$

- Apply the MDS algorithm to a distance or similarity matrix other than geodesic

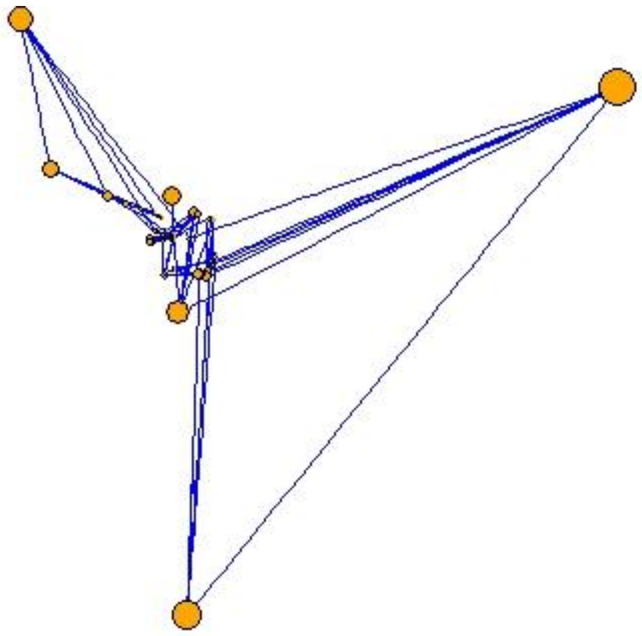
(for example, similarity in terms of structural equivalence)

MDS for social networks

MDS on Adjacency Matrix



MDS on Structural equivalence



Correspondence Analysis

Correspondence Analysis (CA) is a statistical method used to analyze and represent the association structure present in a contingency table.

- Let $\mathbf{N}$ a contingency table $I \times J$ rows for columns of elements n_{ij};
- Let $\mathbf{P}$ the contingency table of elements n_{ij} / n of the relative frequencies;
- Let $\mathbf{S}$ be the standardized residue table under the hypothesis of independence

$$s_{ij} = \left(p_{ij} - r_i c_j\right) \Big/ \sqrt{r_i c_j}$$

AC consists of decomposition in singular values of matrix $\mathbf{S}$

$$\mathbf{S} = \mathbf{D_r}^{-\frac{1}{2}} \left(\mathbf{P} - \mathbf{rc}^{\mathbf{T}}\right) \mathbf{D_C}^{-\frac{1}{2}} = \mathbf{U \Lambda \Lambda}$$

Where $\mathbf{U}$ and $\mathbf{V}$ are the matrices of the right and left singular vectors and $\mathbf{\Lambda}$ is the diagonal matrix of the singular values

CA and networks

Correspondence analysis has been widely used in 2-mode networks analysis.

This method is specifically indicated in the case of weighted networks in which the link represents the number of times the actor participated in the event, or the number of times an author mentions a magazine, or in all those cases in which the value of the tie is a positive number of count.

CA was also used to analyze affiliation matrices, that is, those where the link is binary (0/1), raising some criticisms.

For the mathematical structure of CA unrelated actors or events (that is, with a low degree) weigh heavily (sometimes excessively) in the analysis by obscuring the main structures.

CA and networks: reading criteria

The reading criteria are different from those of nodes-arcs representations.

The CA highlights the similarity among the actors or among the events and their association in terms of structural similarity.

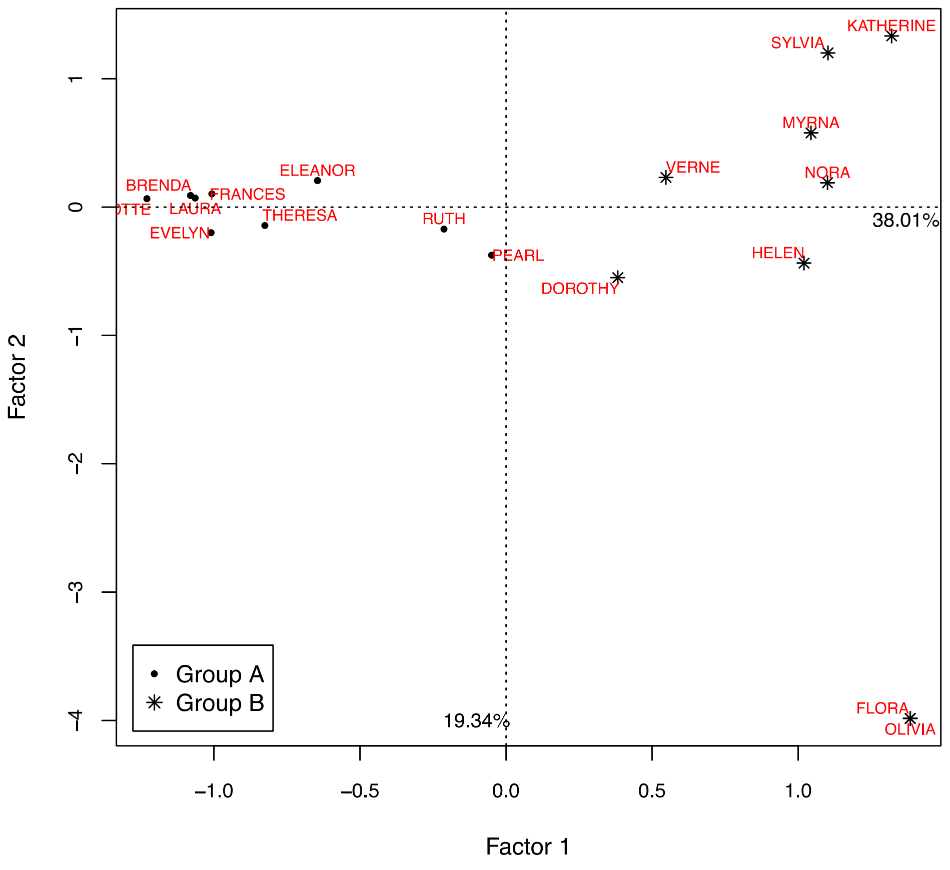
Two actors close to the factorial plane are structurally similar, i.e. participate in the same events.

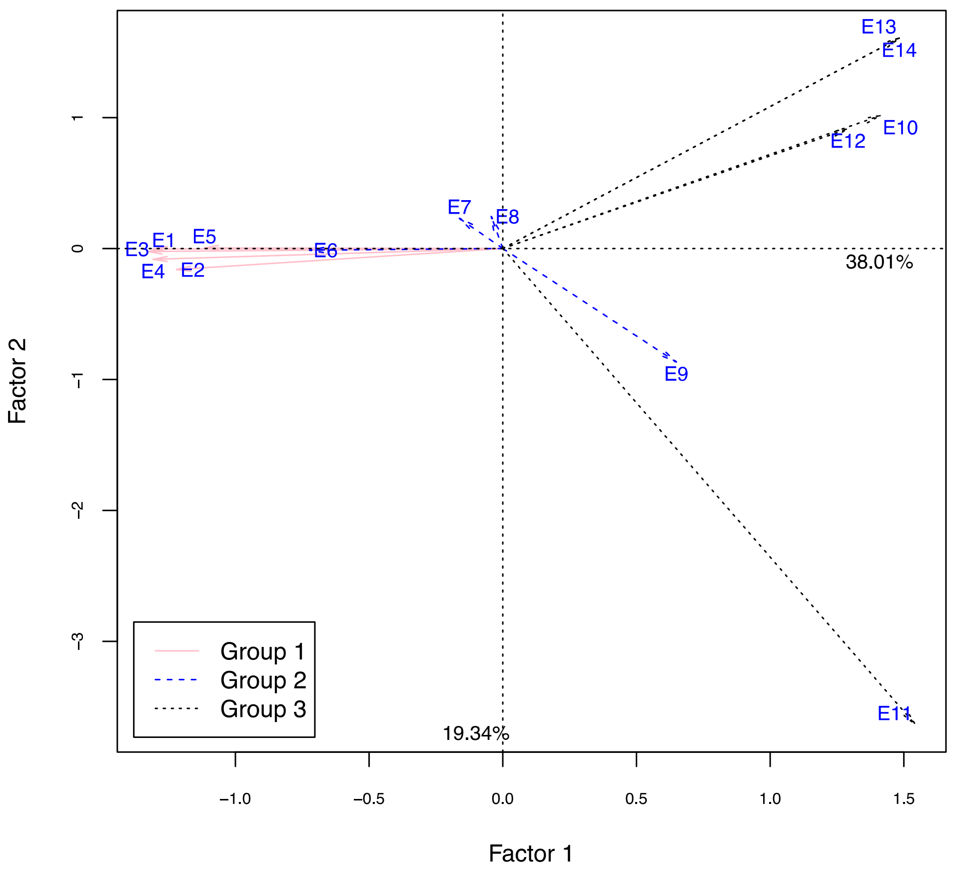
Two events close on the factorial plane are structurally similar, i.e. the same actors participate.

It is possible to represent jointly actors and events.

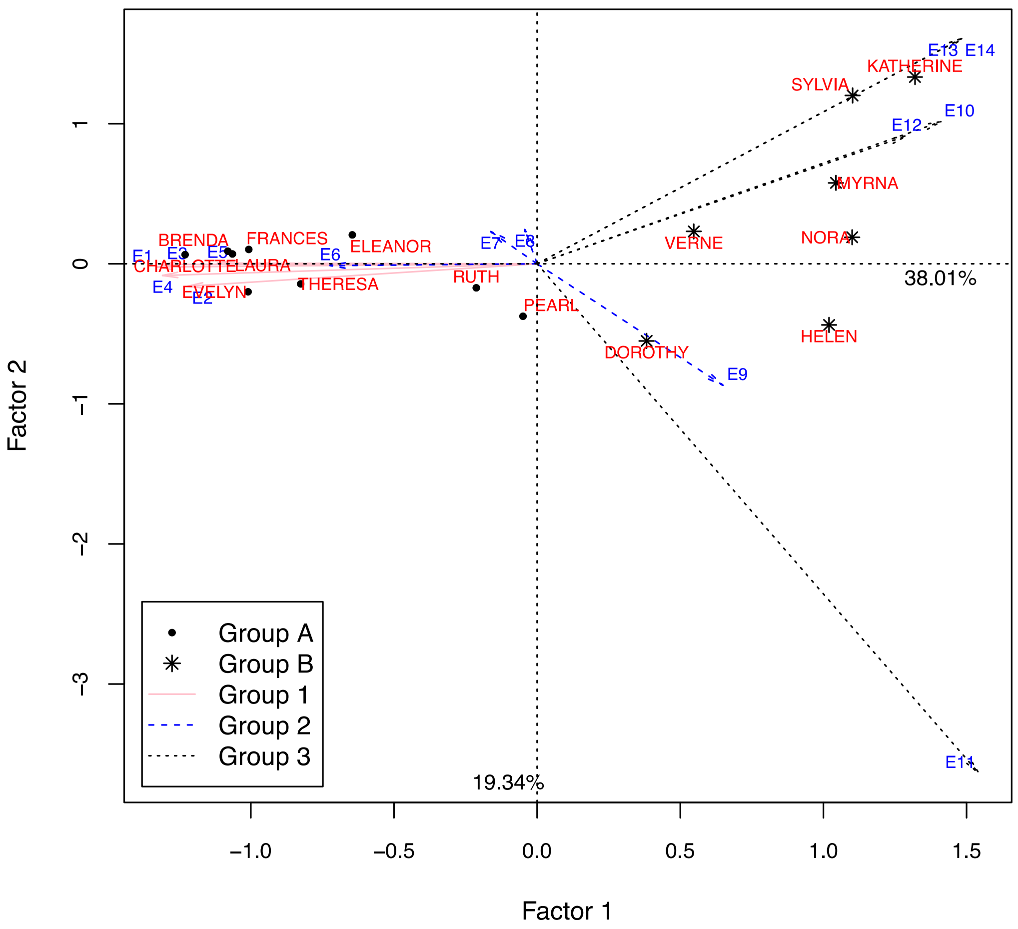
The relation between actors and events should be read in terms of angles and not distance.

CA: reading criteria





CA and networks: reading criteria



Multiple Correspondence Analysis

ELSEVIER

Social Networks

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

On the use of Multiple Correspondence Analysis to visually explore affiliation networks

Maria Rosaria D'Esposito[a], Domenico De Stefano[b,*], Giancarlo Ragozini[c]

Multiple Correspondence Analysis (MCA) is a statistical method used to analyze and represent the association structure present in several qualitative variables.

There are numerous formulations. We refer to the case in which there are individuals (actors) on the rows and categorical variables on the columns (participation in events)

- Let $\mathbf{F}$ a matrix that encodes the affiliation matrix (we treat it as individuals for variables);
- Let $\mathbf{Z}$ the table in complete disjunctive coding $\mathbf{Z} = [\mathbf{F}^{+} \mid \mathbf{F}^{-}]$
- Let $\mathbf{S}$ the standardized residue table under the hypothesis of independence will be

$$\mathbf{S} = \mathbf{D_a}^{-1/2}\left(\frac{\mathbf{Z}}{nm} - \mathbf{D_a}\mathbf{1}\mathbf{1}^T\mathbf{D_e}\right)\mathbf{D_e}^{-1/2} = \sqrt{n}\left(\frac{\mathbf{Z}}{nm} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{D_e}\right)\mathbf{D_e}^{-1/2},$$

- MCA is the decomposition in singular values of $\mathbf{S}$

	e_1	e_2	e_3	e_4
a_1	1	0	0	1
a_2	1	0	1	0
a_3	0	1	1	1

F- Affiliation matrix

	e_1		e_2		e_3		e_4	
	+	−	+	−	+	−	+	−
a_1	1	0	0	1	0	1	1	0
a_2	1	0	0	1	1	0	0	1
a_3	0	1	1	0	1	0	1	0

Z- Full disjunctive coding of F

Fig. 1. A fictitious affiliation matrix **F** and the corresponding indicator matrix **Z** obtained through full disjunctive coding.

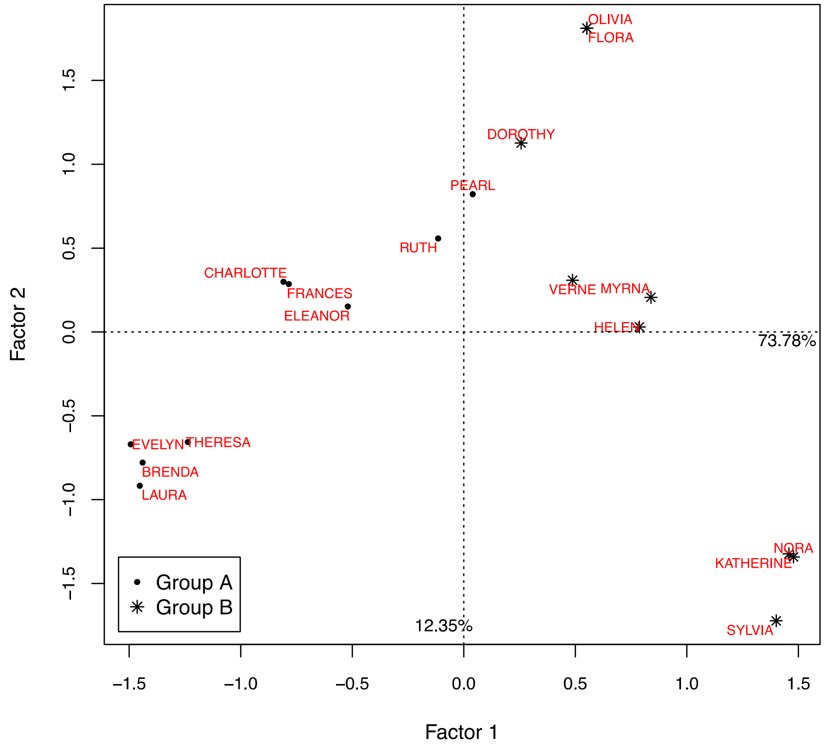
Multiple Correspondence Analysis

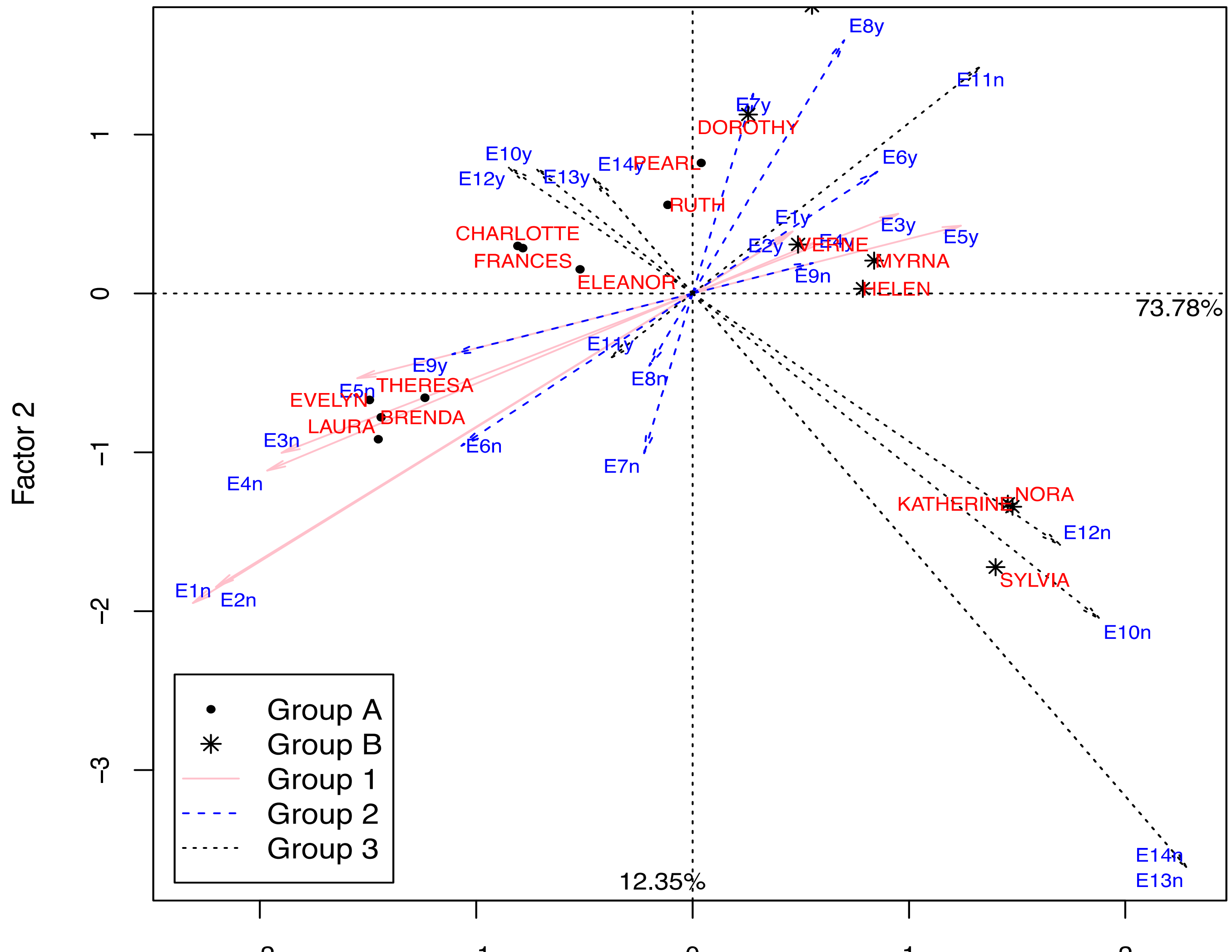
Social Networks

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

On the use of Multiple Correspondence Analysis to visually explore affiliation networks

Maria Rosaria D'Esposito[a], Domenico De Stefano[b,*], Giancarlo Ragozini[c]

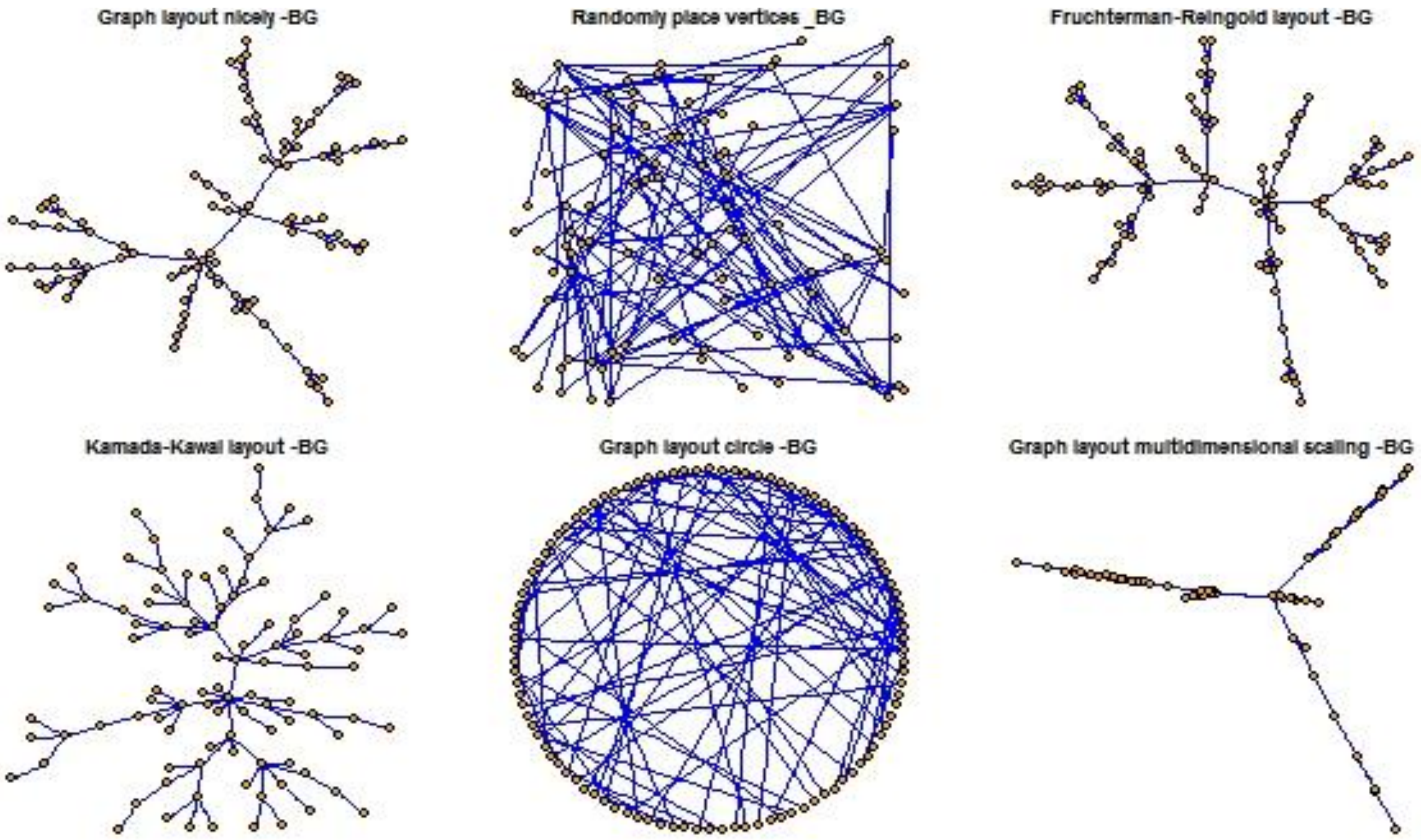




Nodes-ties structural representations

DIFFERENT LAYOUTS

Some comparisons

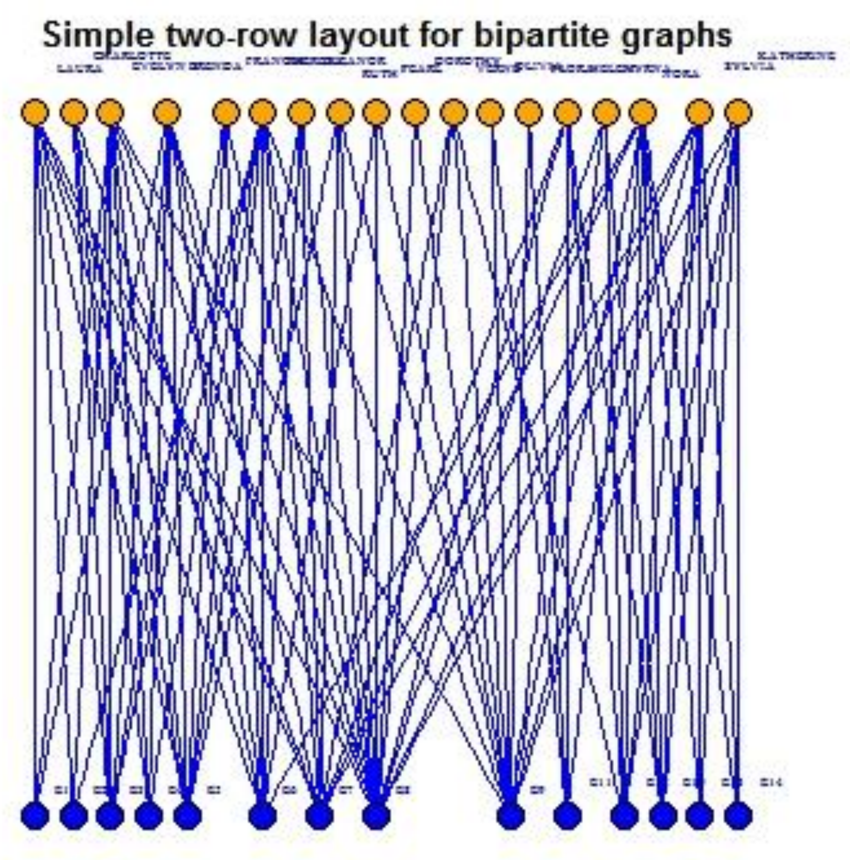


Bipartite Graph

The bipartite graph is the simplest layout for 2-mode networks.

The two modes of the network are represented with different symbols and colors aligned one face to the other

Arcs or segments from one mode to another connect the nodes of the two sets.



Visual alphabet of nodes-ties representations

- Dimensions
- Forms and simbols
- Lines
- Colors

Visual alphabet of nodes-ties representations

Dimensions

- The size of the nodes is one of the main stimuli of node-tie representation.
- Too small nodes or too large nodes can produce misleading representations.

Visual alphabet of nodes-ties representations

Forms and Symbols

- The basic shape for the node is the circle or the sphere, but many other shapes and symbols can be used.
- Is relevant that they are meaningful, that is consistent: for example homogeneous groups must be represented by similar symbols.

Visual alphabet of nodes-ties representations

Lines

- It is possible to choose whether the ties are segments, curves, arcs.
- It is possible to change the thickness of the lines, the color of the lines.
- For large and dense networks it is good to use transparency.

Visual alphabet of nodes-ties representations

Colors

- The colors must create a distinguishable and orderly visualization.
- The colors have an aesthetic value, but they also depend on culture and can generate different psychological reactions.
- Tone, brightness and saturation.

Nodes-ties structural representations

INCORPORATE GRAPH PROPERTIES

Show attributes and roles and positions

Dimensions, shapes and symbols, lines and colors can be used to encode additional information within the nodes-ties representation

Examples:

Actors attributes: categorical variables or group memberships can be encoded with different shapes, symbols, or colors.

Centralization measures: Continuous variables can be related to the size of the symbols or the color saturation level.

Ties attributes: can be encoded with different line or with different colors.

Transparency and saturation reveals the dense and sparse areas of the graph.

SOCIOMATRICES
Representation

Sociomatrices representation

An alternative way of representing social networks such as graphs is the direct representation of sociomatices.

The matrix is represented as a square or rectangle and divided into as many squares as the cells of the sociomatrix.

Each square is filled if the tie is present (if the network is weighed it is possible to fill the square with a color of varying intensity as the weight changes).

Even in this case the basic problem is sorting rows and columns, which in general is a problem

$NP\text{-}Hard$.

The main goal is to show clusters and groups and so it is very much used after clustering, community detection, or blockmodeling.

Sociomatrices representation

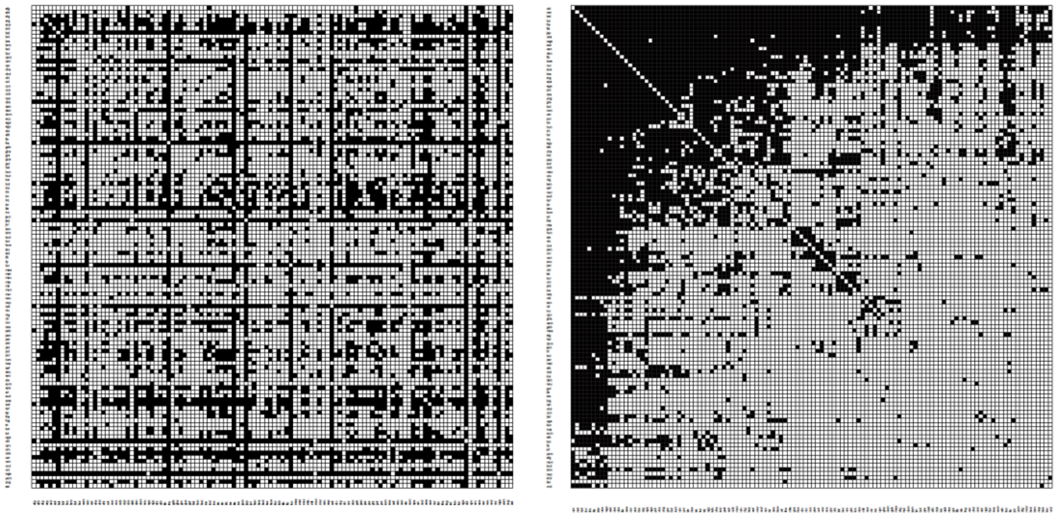
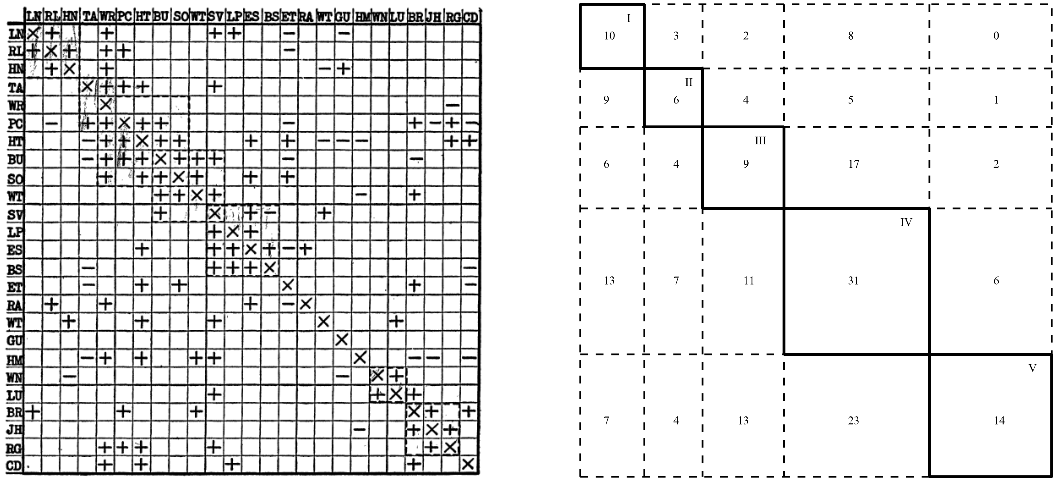


Figure 26.11 Trade between countries reordered according to a hierachical clustering (reproduced from [BM04]).



(a) ordered sociomatrix of a signed graph [FK46]

(b) blocked sociomatrix with edge counts [Lon48]

Figure 26.10 Sociomatrix and block partition.

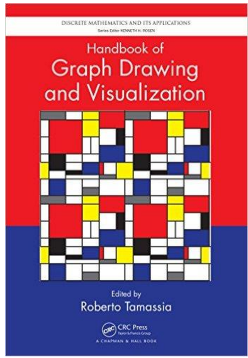
26

Social Networks

Ulrik Brandes
University of Konstanz

Linton C. Freeman
University of California, Irvine

Dorothea Wagner
Karlsruhe Institute of Technology



Sociomatrices representation

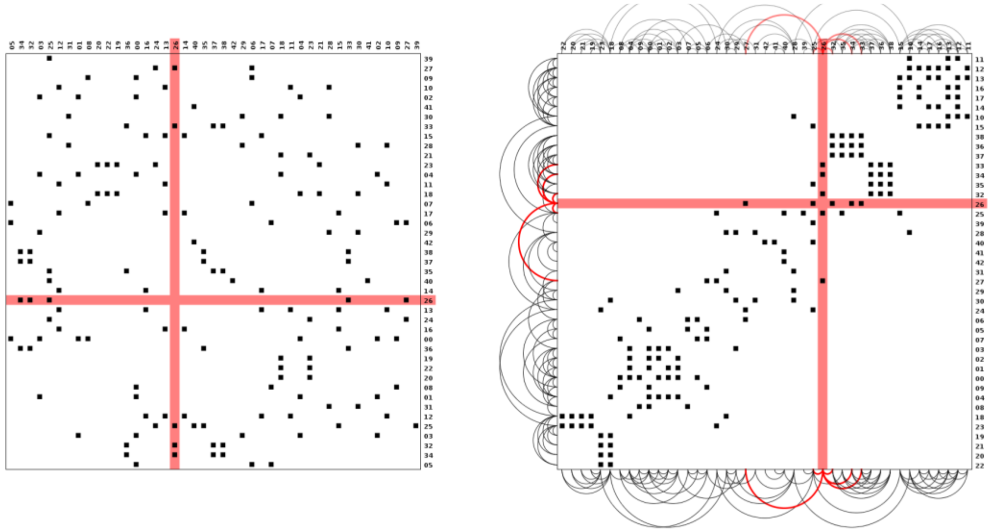


Fig. 5 Adjacency matrix visualizations of a 43-node, 80-edge network. Top: with a random ordering of rows and columns. Bottom: after barycenter ordering and adding arc diagrams. The multiple arc diagrams are redundant, but reduce the distance of eye movements from the inside of the matrix to the nearest arcs.

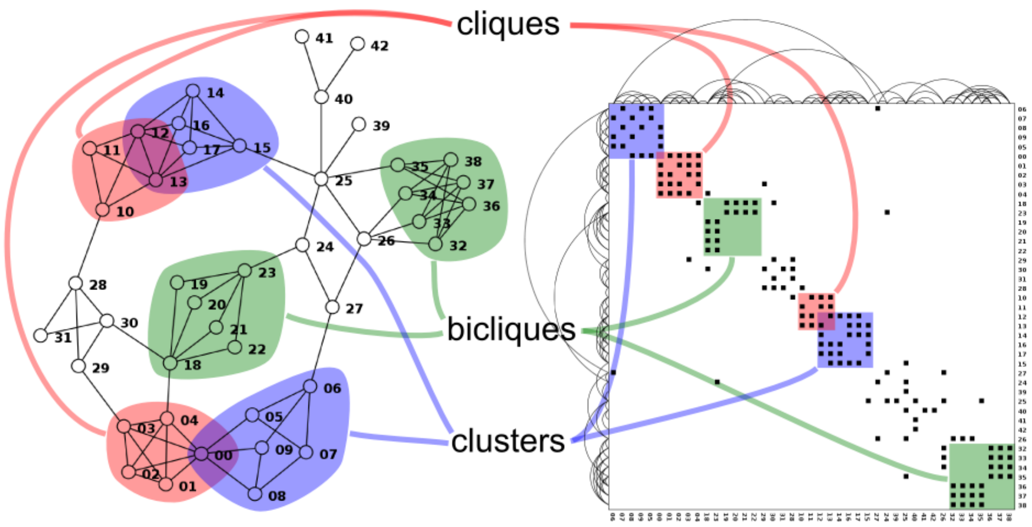


Fig. 6 Patterns corresponding to interesting subgraphs appear along the diagonal of an appropriately ordered adjacency matrix.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/12 pp1-16
Volume 17, Number 4, August 2012

Simple Algorithms for Network Visualization: A Tutorial*

Michael J. McGuffin**

Representations: final consideration

- Many possible layouts
- Many possible aspects to highlight
- Many choices possible for size, shape, color, lines

It follows that there are so many possible representations for the same network and making a good representation is a matter of "artistic craftsmanship" and experience