CS6220: DATA MINING TECHNIQUES

Mining Graph/Network Data: Part II

Instructor: Yizhou Sun

yzsun@ccs.neu.edu

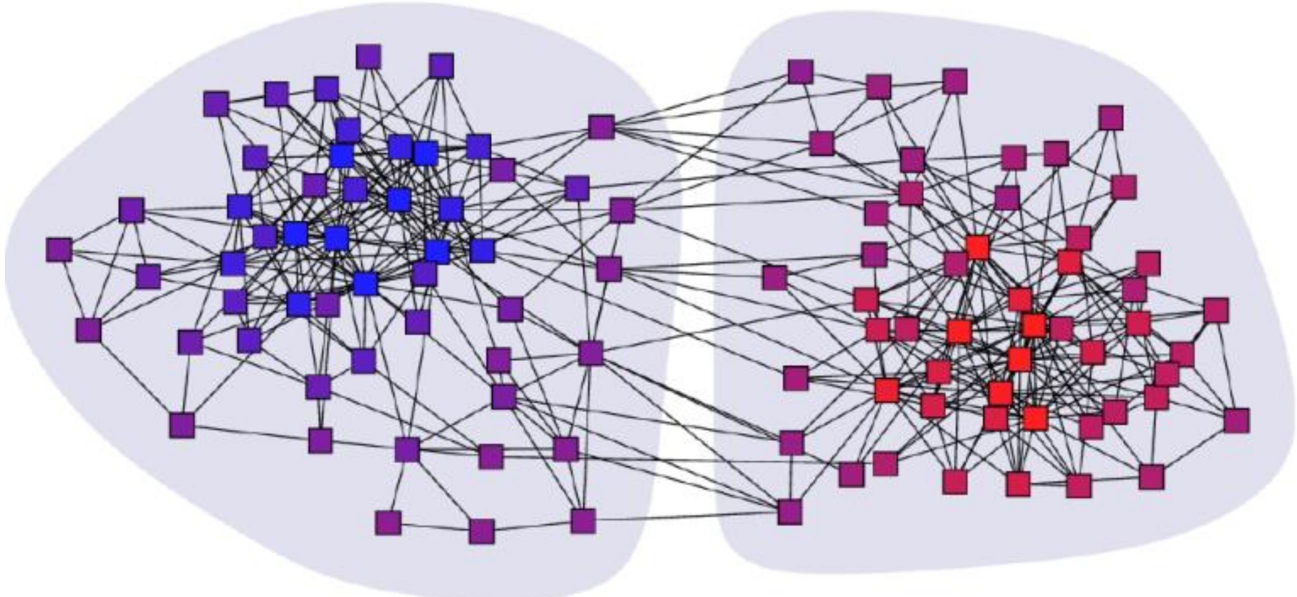
November 26, 2013

Mining Graph/Network Data: Part II

- Graph/Network Clustering
- Graph/Network Classification
- Summary

Clustering Graphs and Network Data

- Applications
  - Bi-partite graphs, e.g., customers and products, authors and conferences
  - Web search engines, e.g., click through graphs and Web graphs
  - Social networks, friendship/coauthor graphs



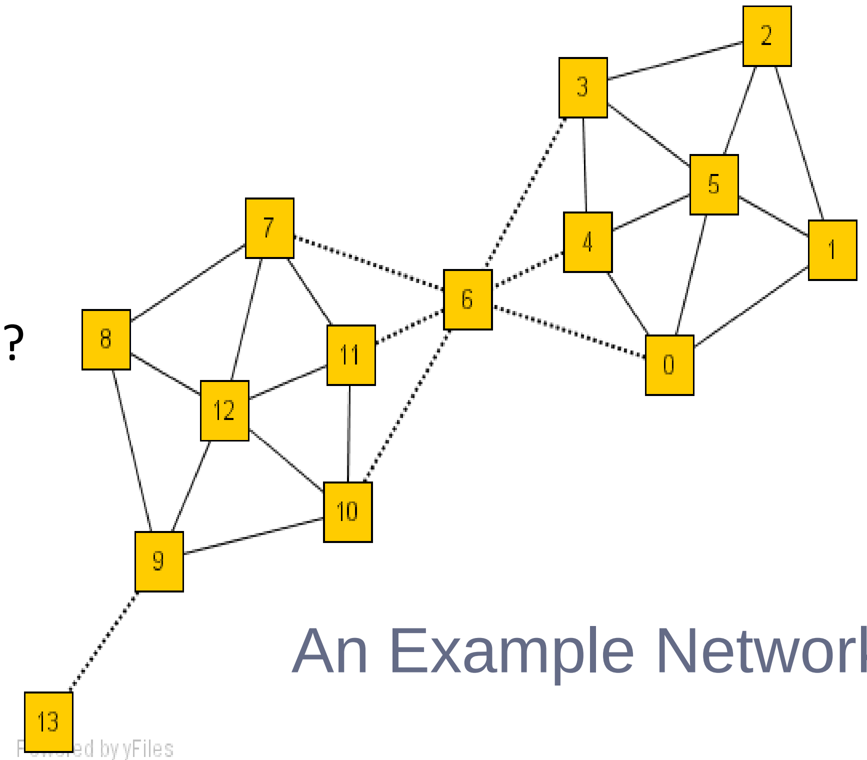
Clustering books about politics [Newman, 2006]

Algorithms

- Graph clustering methods
  - Density-based clustering: SCAN (Xu et al., KDD'2007)
  - Spectral clustering
  - Modularity-based approach
  - Probabilistic approach
  - Nonnegative matrix factorization
  - …

SCAN: Density-Based Clustering of Networks

- How many clusters?
- What size should they be?
- What is the best partitioning?
- Should some points be segregated?

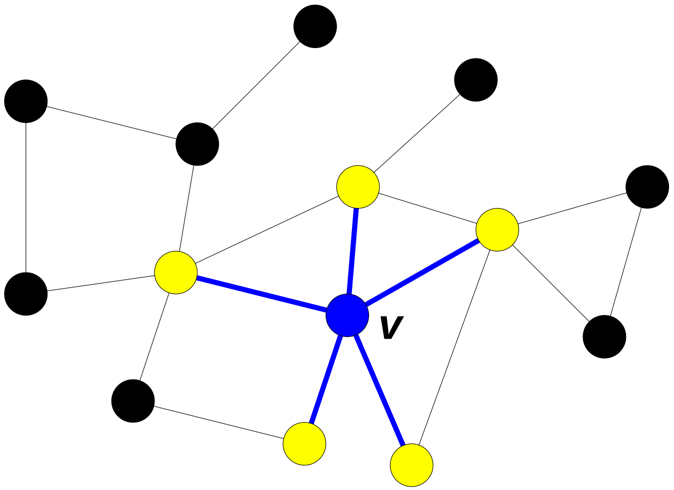


An Example Network

- Application: Given simply information of who associates with whom, could one identify clusters of individuals with common interests or special relationships (families, cliques, terrorist cells)?

A Social Network Model

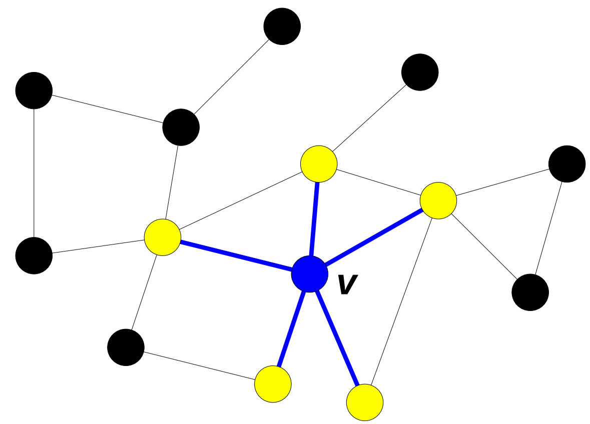
- Cliques, hubs and outliers
  - Individuals in a tight social group, or clique, know many of the same people, regardless of the size of the group
  - Individuals who are hubs know many people in different groups but belong to no single group. Politicians, for example bridge multiple groups
  - Individuals who are outliers reside at the margins of society. Hermits, for example, know few people and belong to no group
- The Neighborhood of a Vertex
  - Define $\Gamma(\nu)$ as the immediate neighborhood of a vertex (i.e. the set of people that an individual knows)



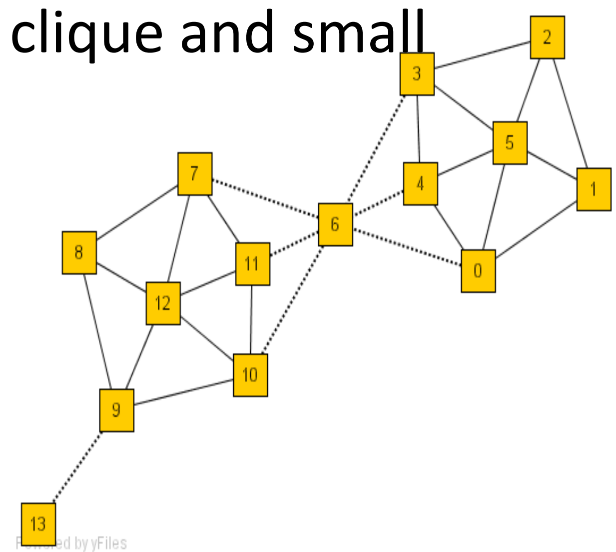
Structure Similarity

- The desired features tend to be captured by a measure we call Structural Similarity

$$\sigma(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$



- Structural similarity is large for members of a clique and small for hubs and outliers



Structural Connectivity [1]

- ε-Neighborhood: $N_{\varepsilon}(v) = \{w \in \Gamma(v) \mid \sigma(v,w) \geq \varepsilon\}$
- Core: $CORE_{\varepsilon,\mu}(v) \Leftrightarrow |N_{\varepsilon}(v)| \geq \mu$
- Direct structure reachable:

$$DirRECH_{\varepsilon,\mu}(v,w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \wedge w \in N_{\varepsilon}(v)$$

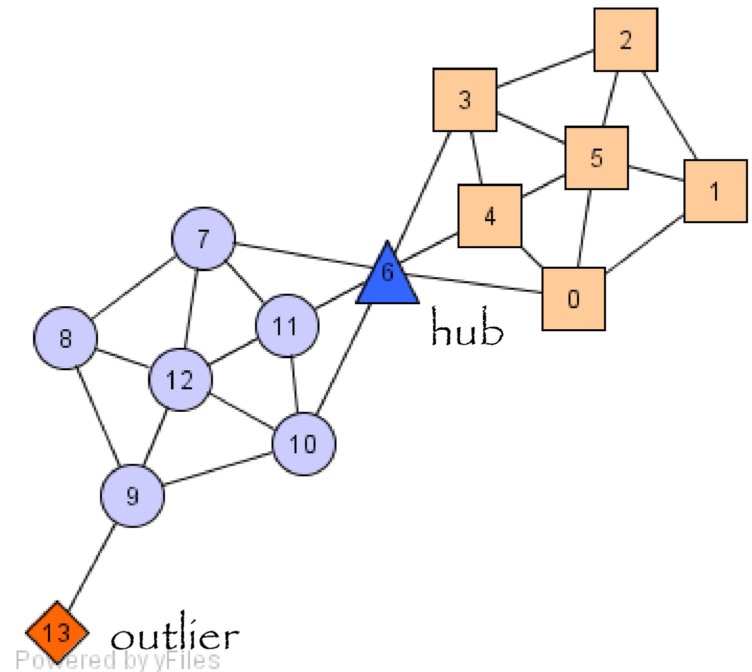
- Structure reachable: transitive closure of direct structure reachability
- Structure connected:

$$CONNECT_{\varepsilon,\mu}(v,w) \Leftrightarrow \exists u \in V : RECH_{\varepsilon,\mu}(u,v) \wedge RECH_{\varepsilon,\mu}(u,w)$$

[1] M. Ester, H. P. Kriegel, J. Sander, & X. Xu (KDD'96) “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases

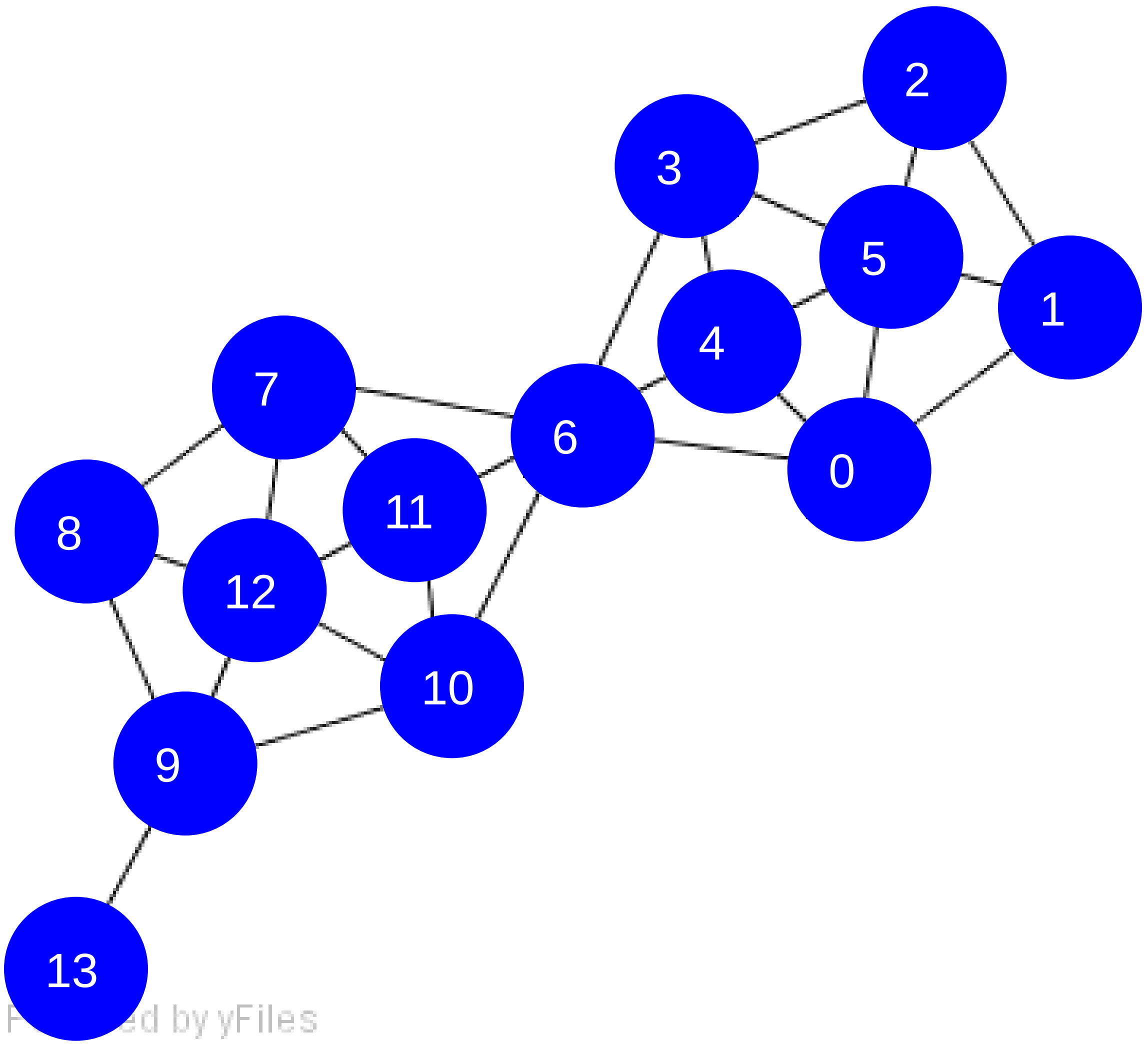
Structure-Connected Clusters

- Structure-connected cluster C
  - Connectivity: $\forall v, w \in C : CONNECT_{\varepsilon,\mu}(v, w)$
  - Maximality: $\forall v, w \in V : v \in C \wedge REACH_{\varepsilon,\mu}(v, w) \Rightarrow w \in C$
- Hubs:
  - Not belong to any cluster
  - Bridge to many clusters
- Outliers:
  - Not belong to any cluster
  - Connect to less clusters



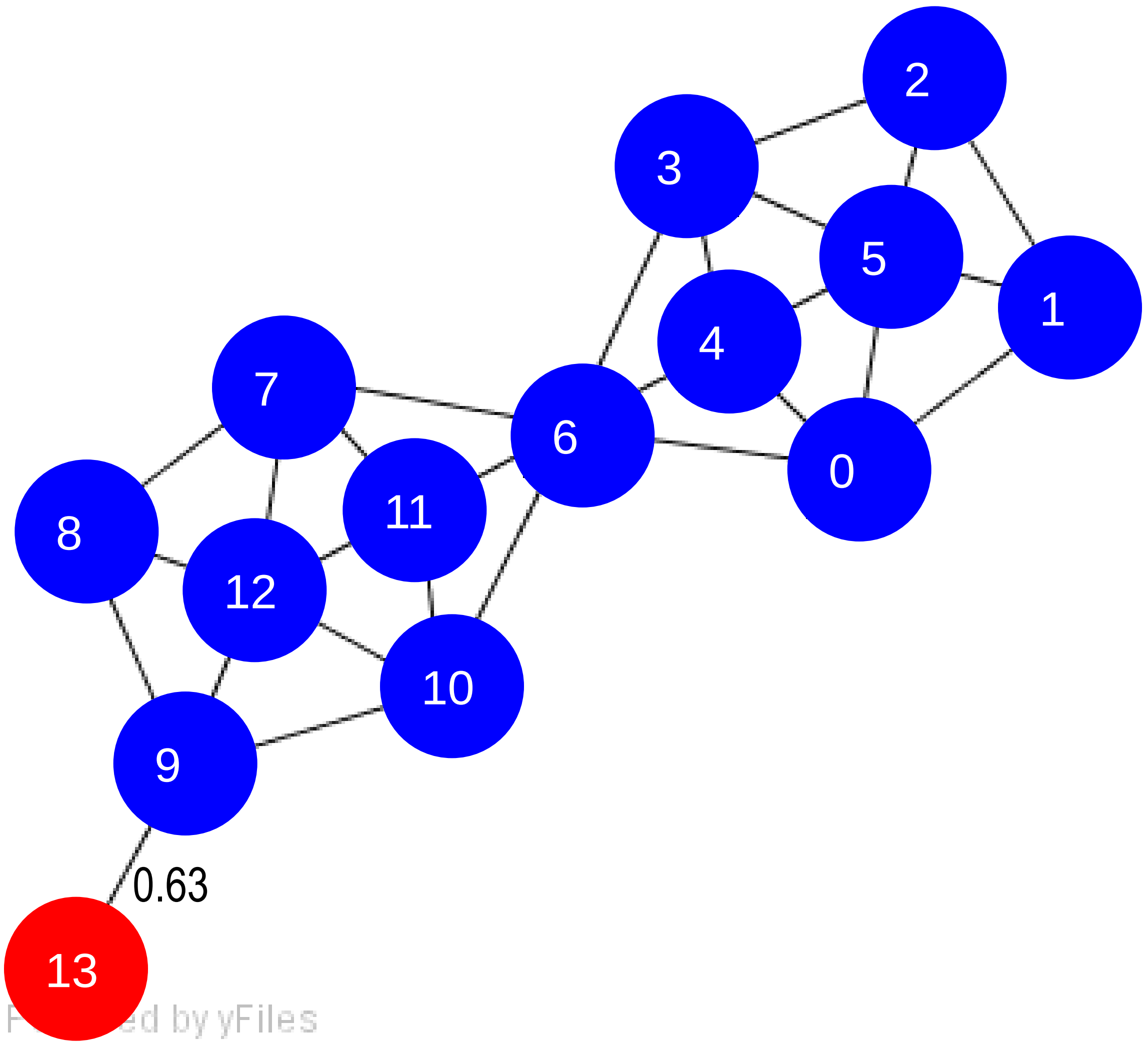
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



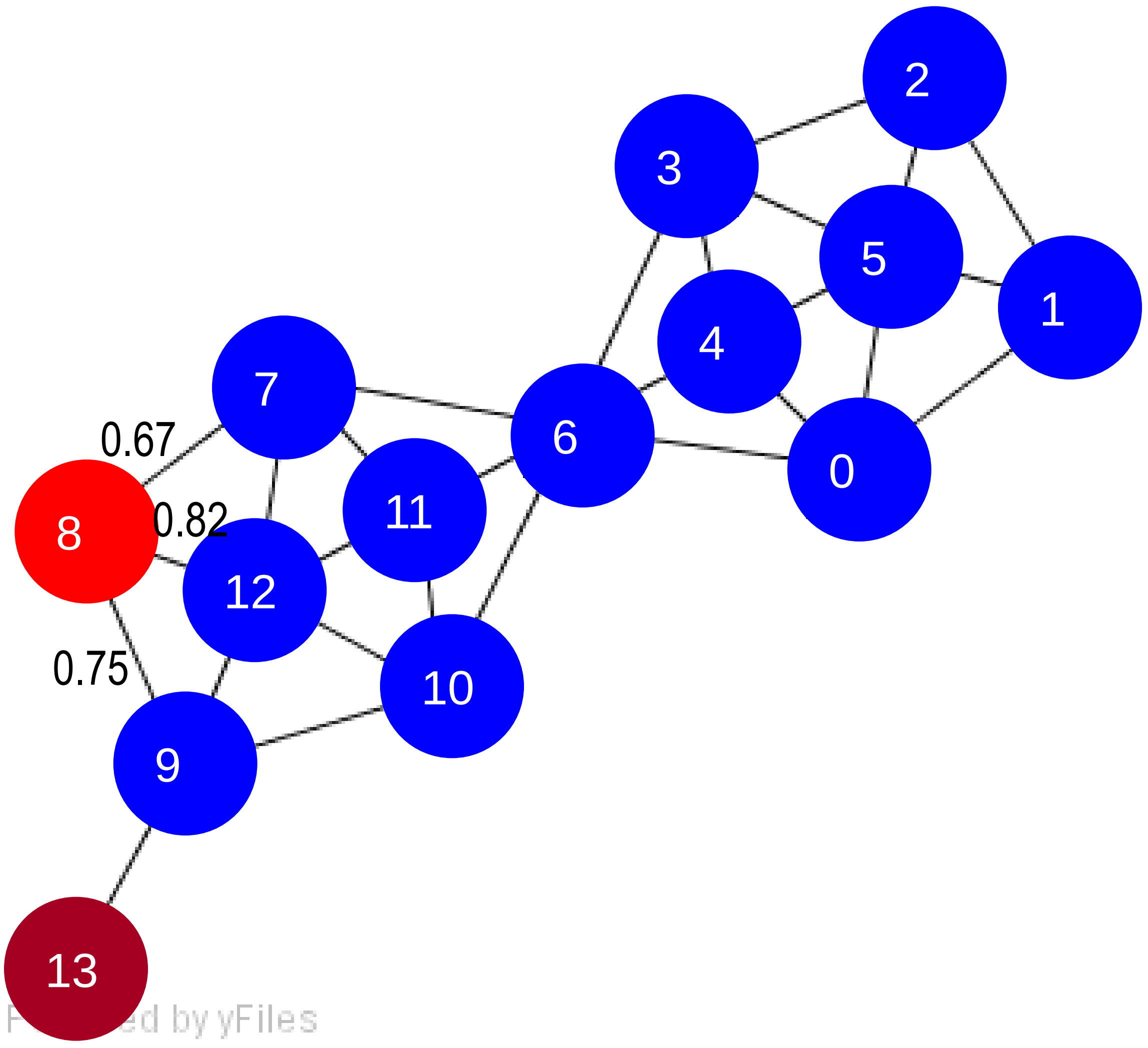
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



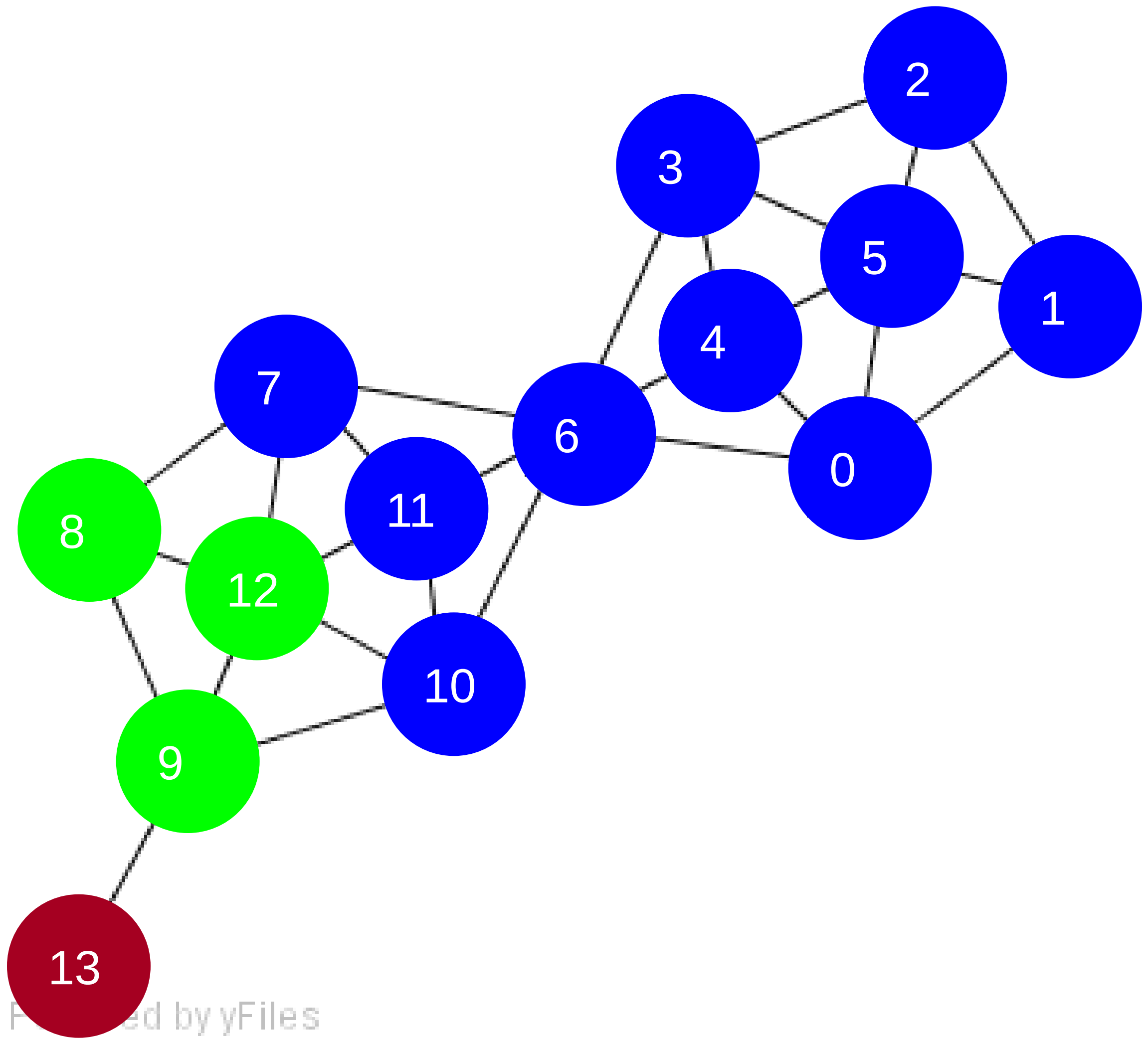
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



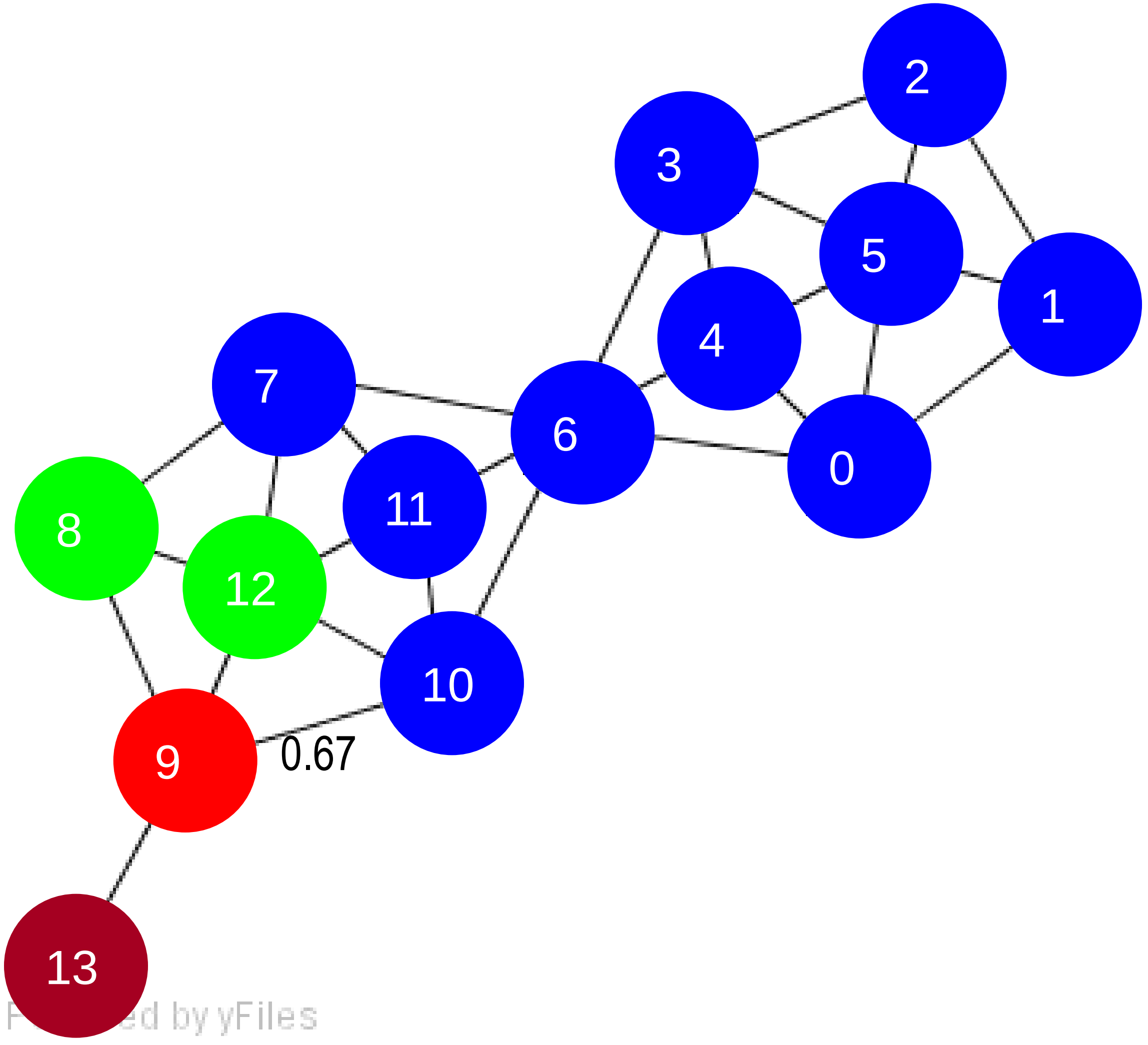
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



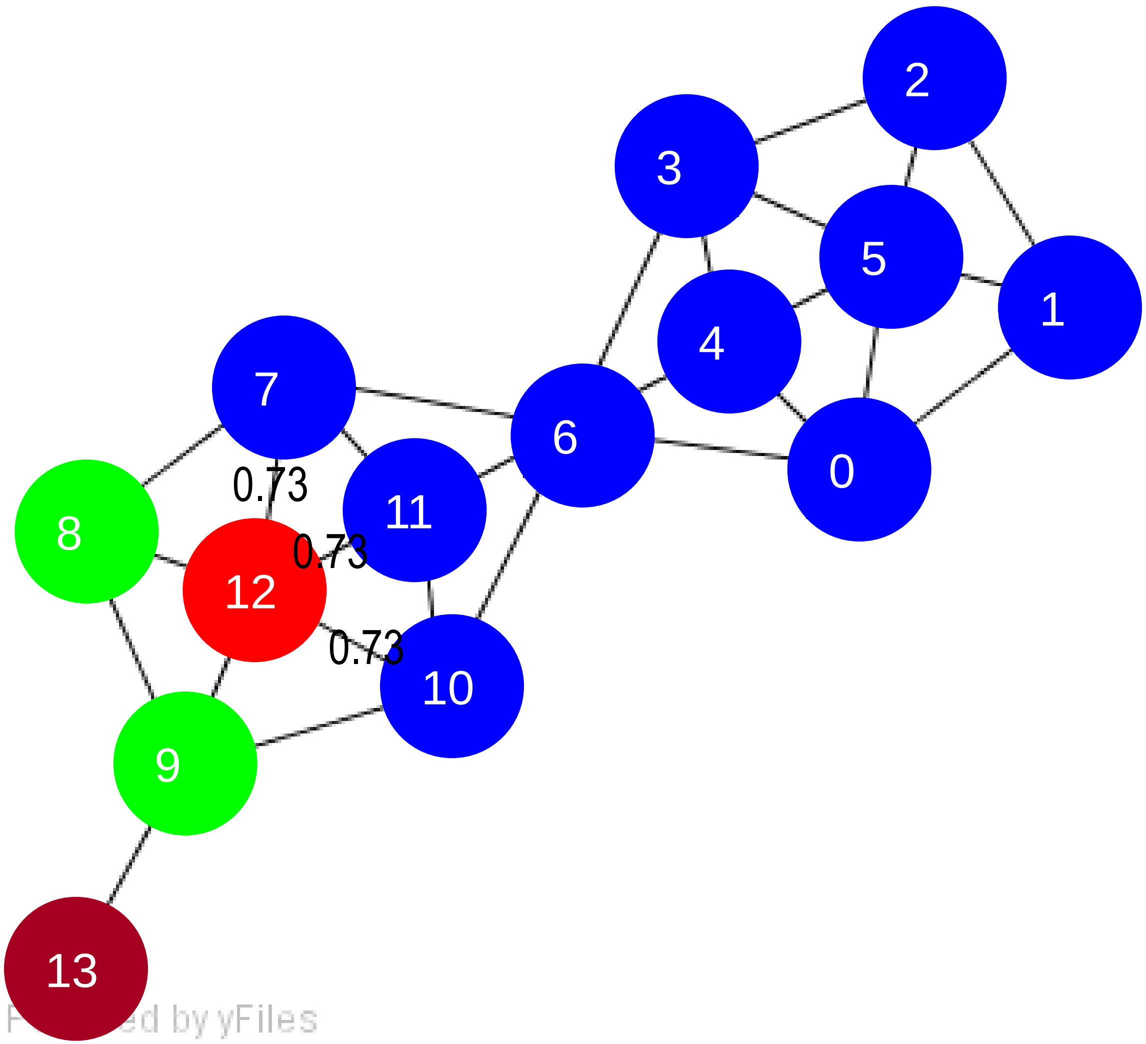
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



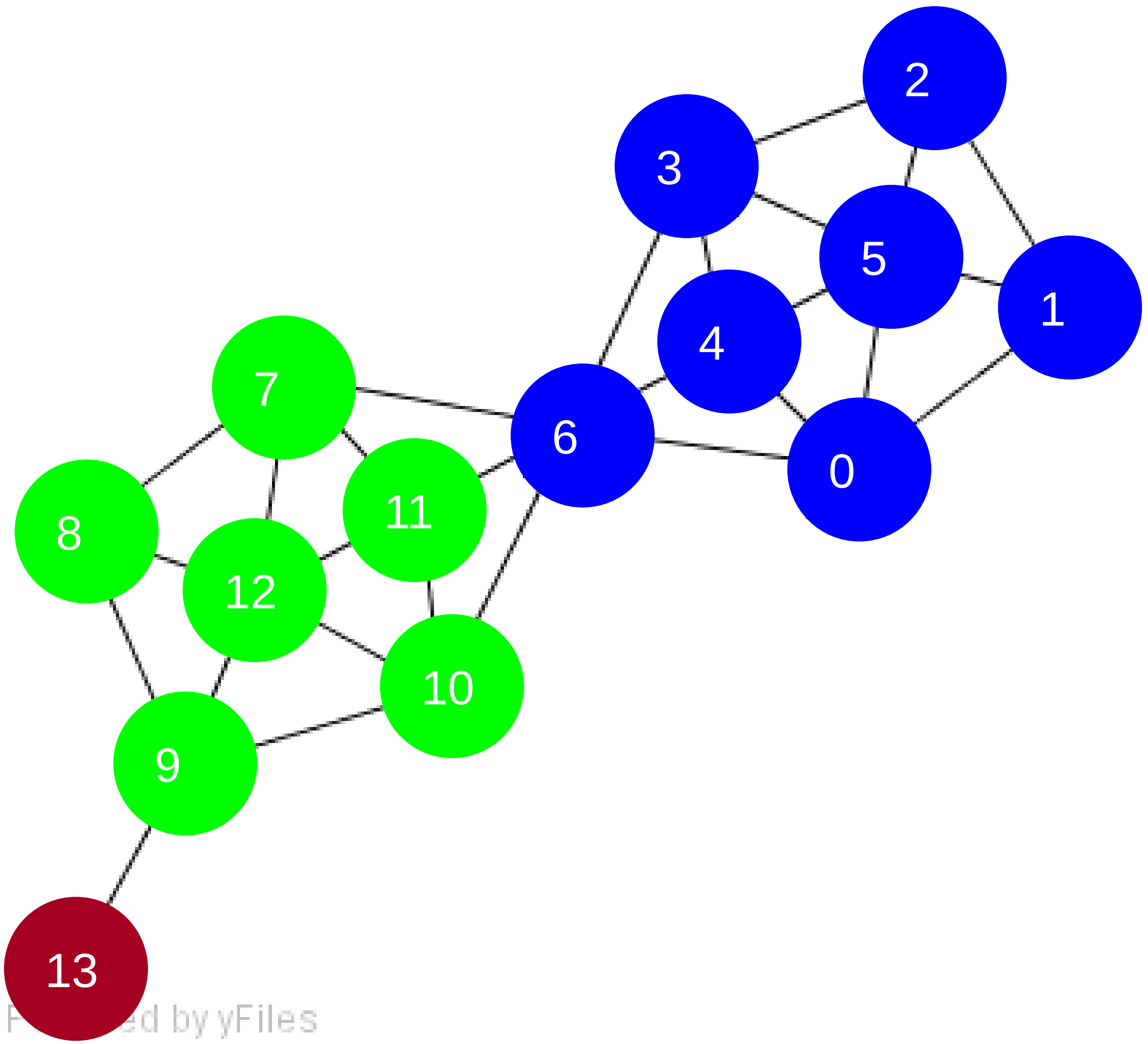
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



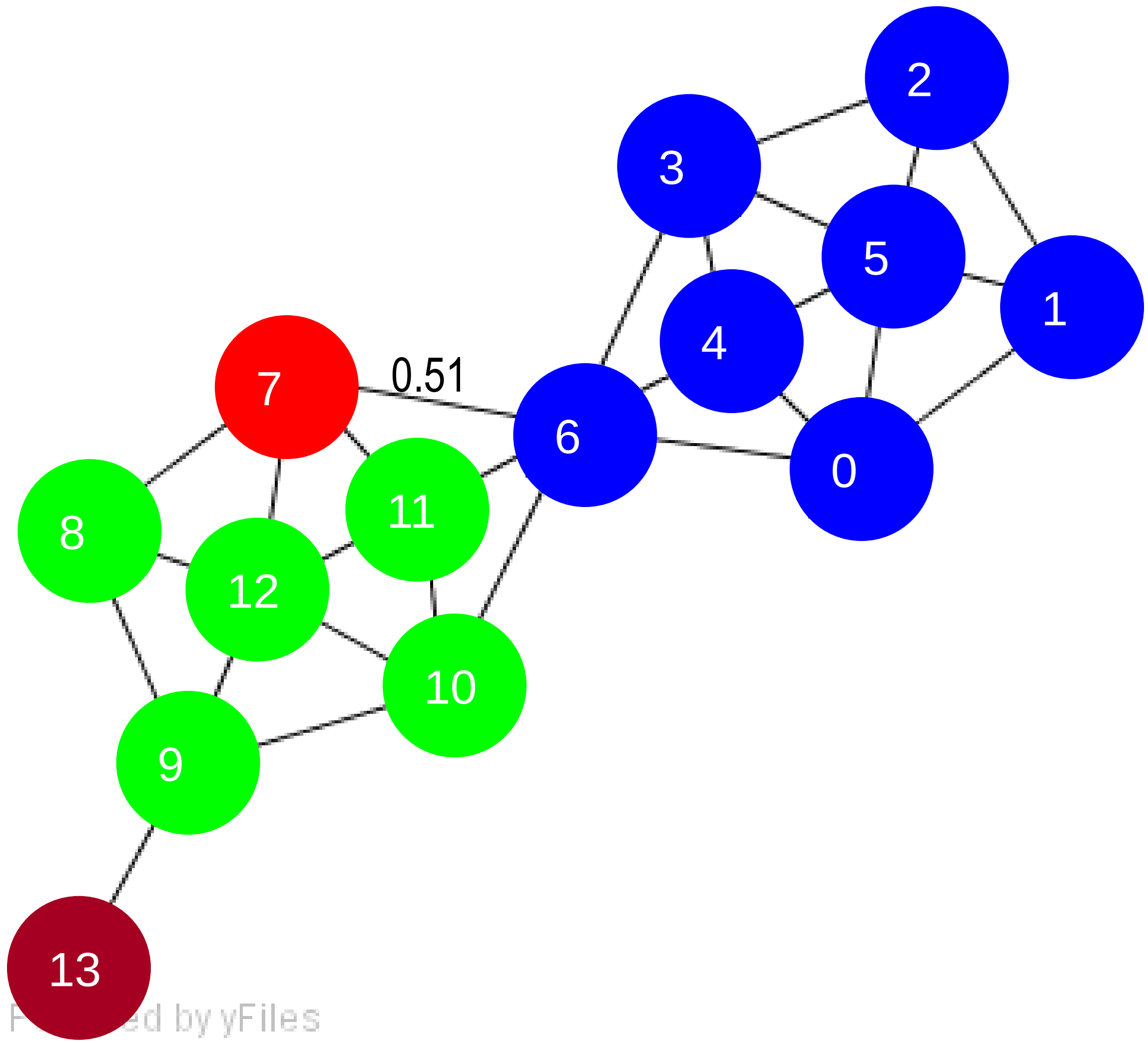
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



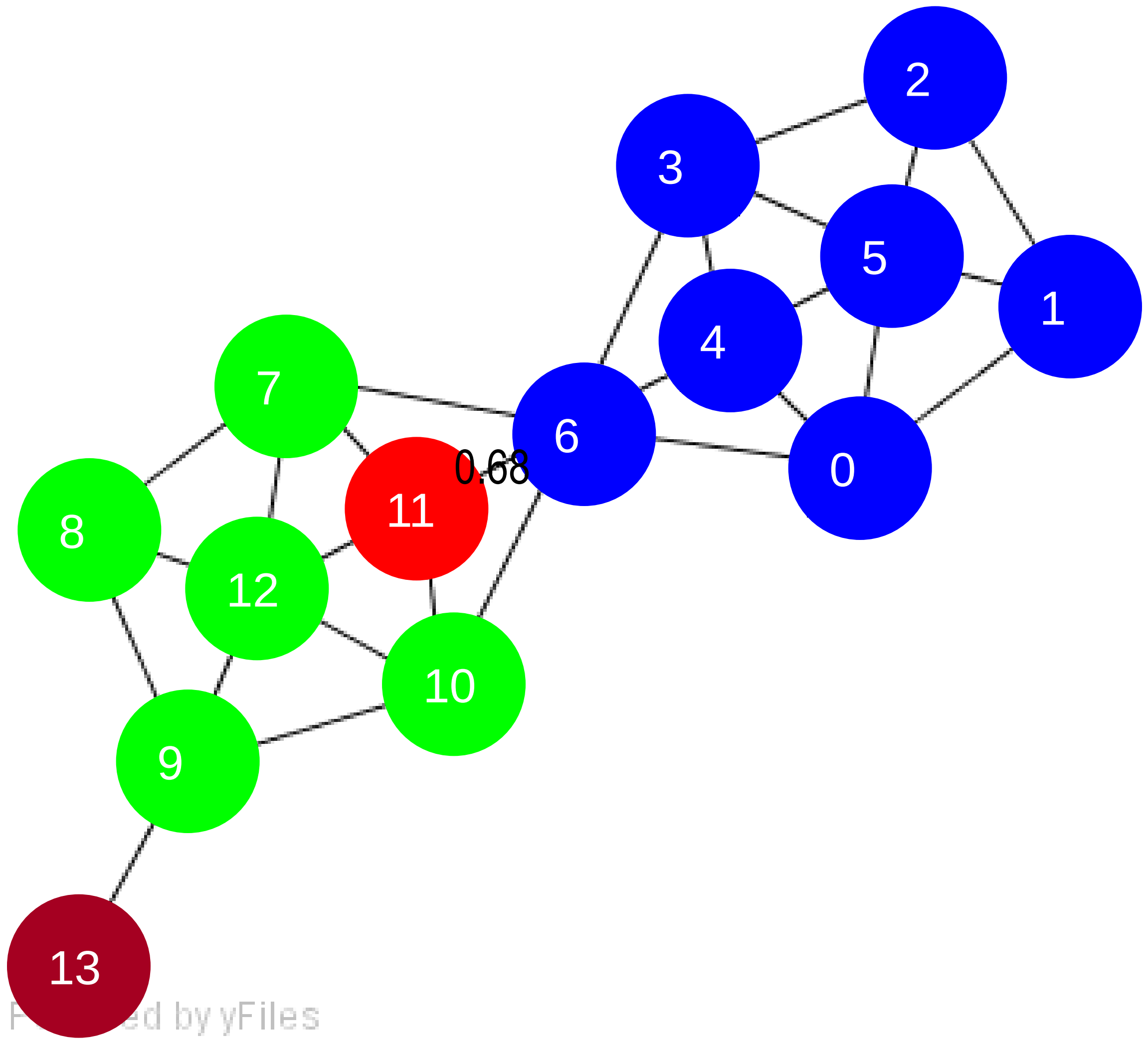
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



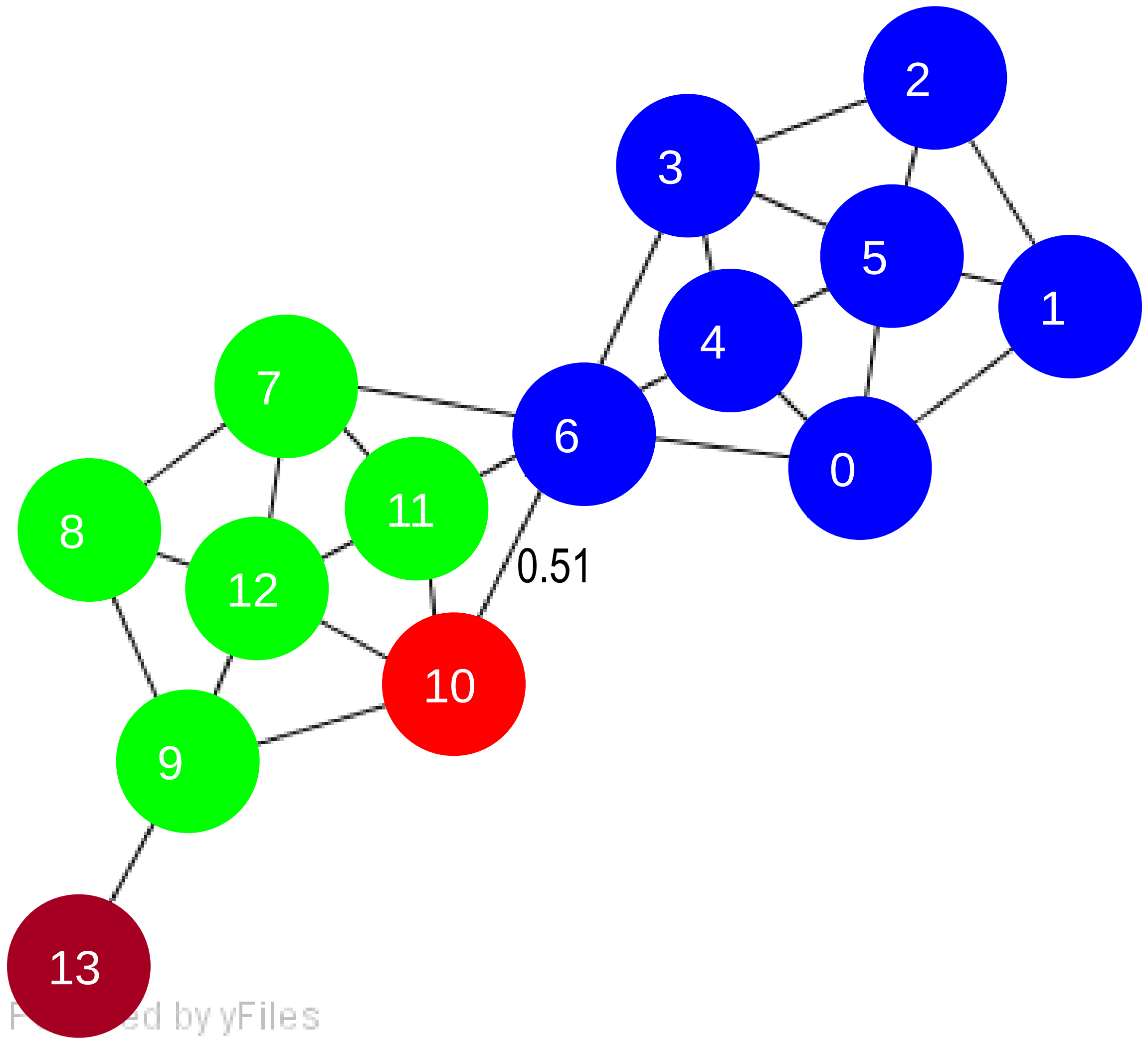
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



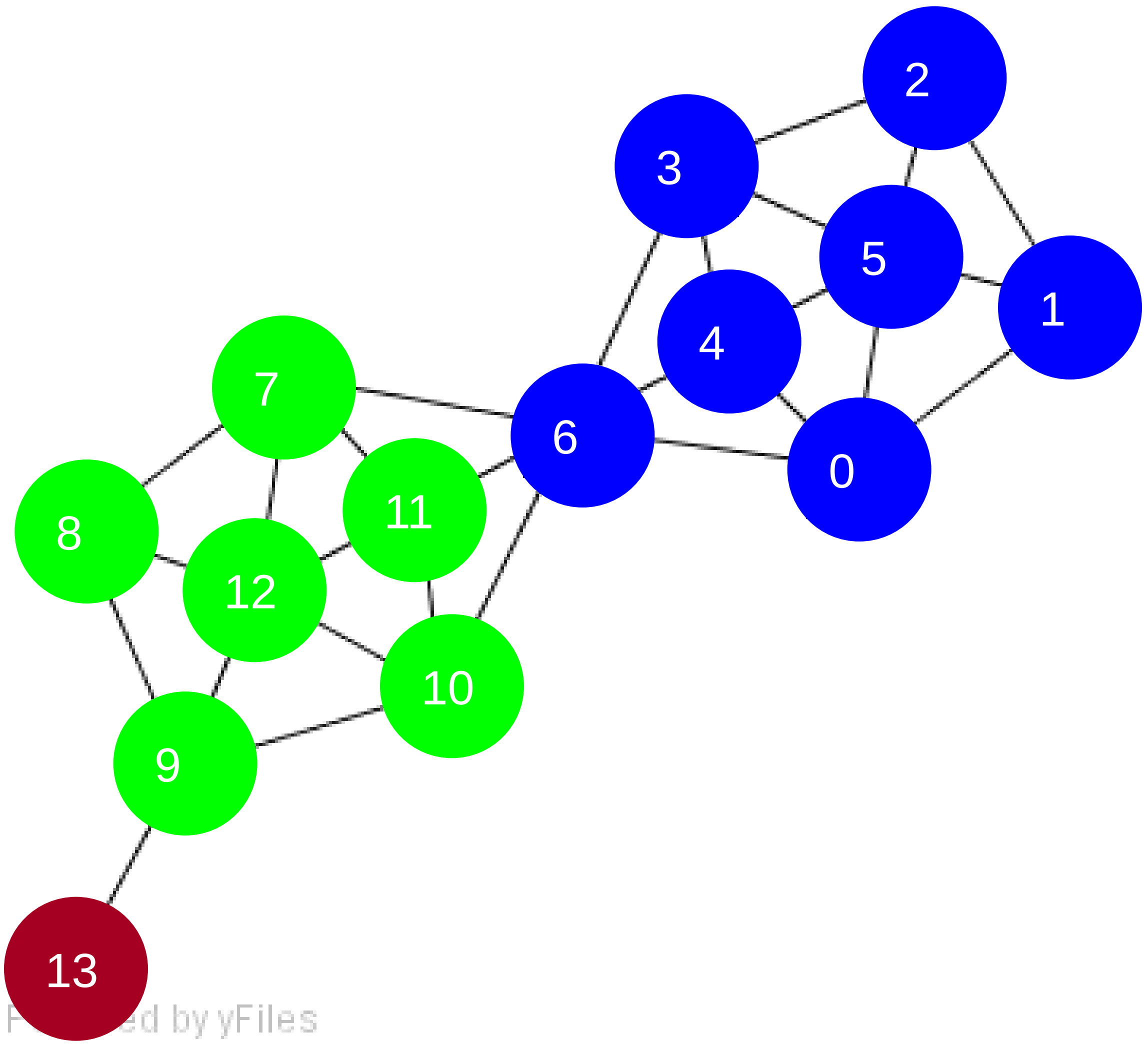
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



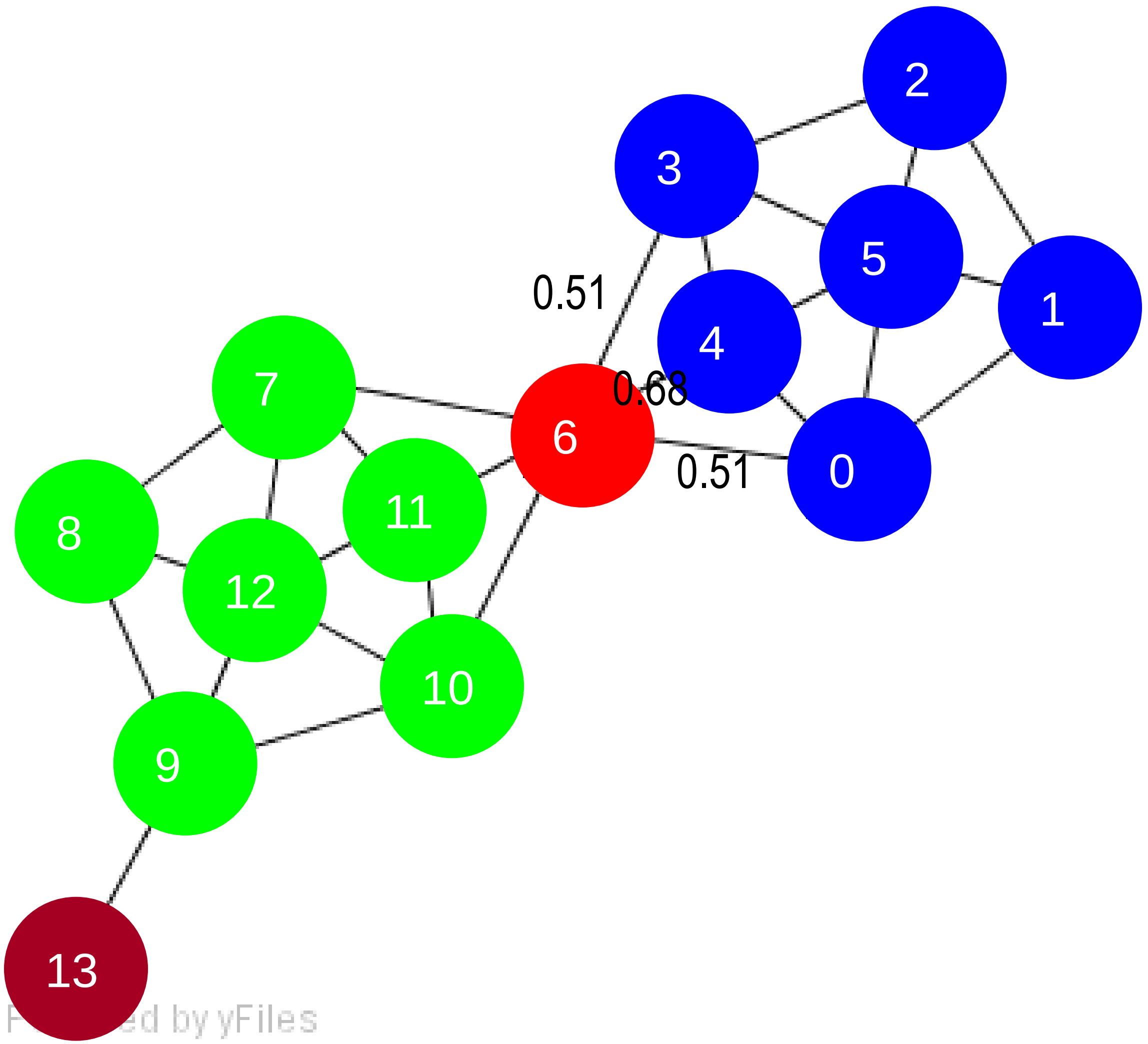
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



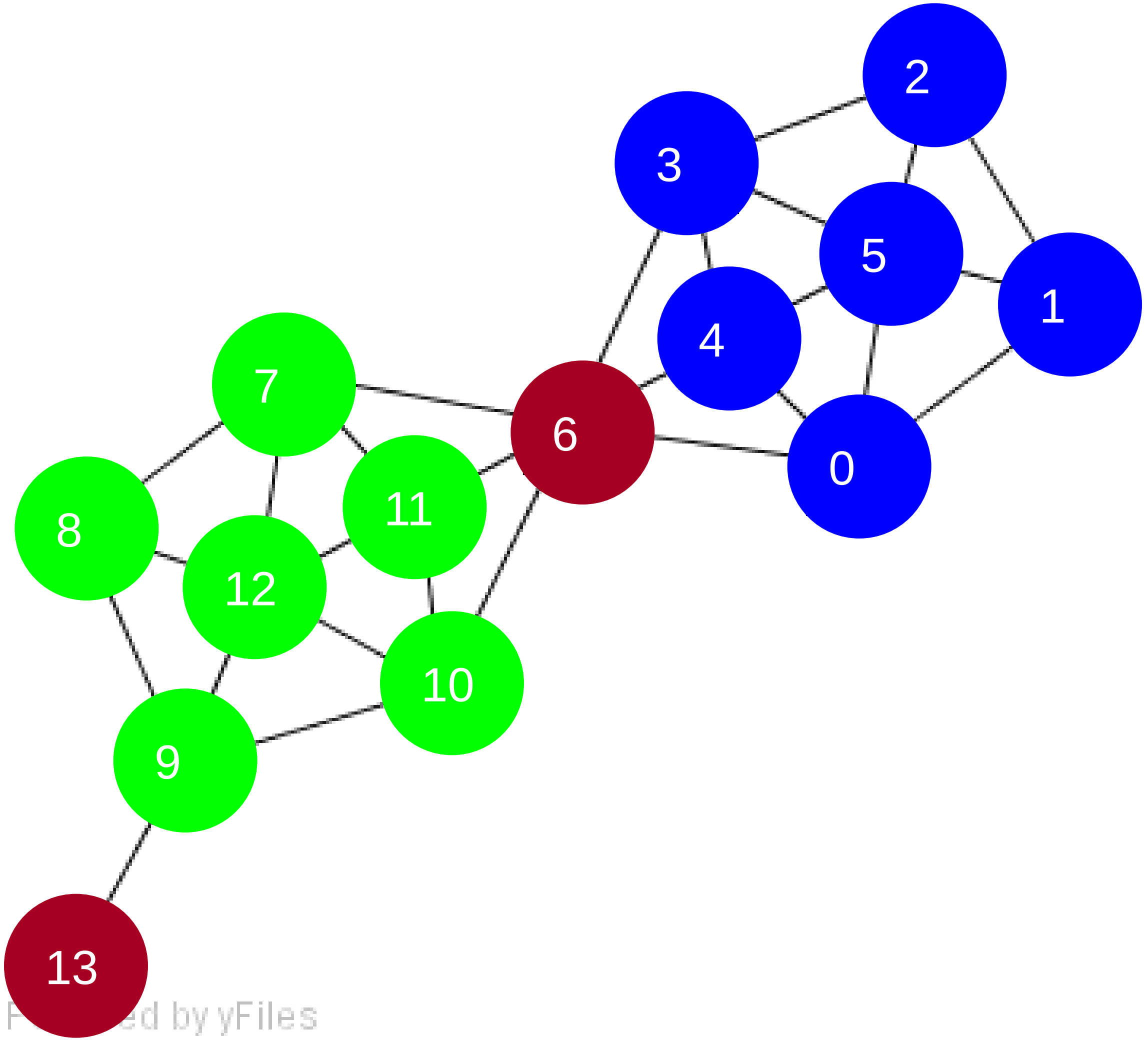
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



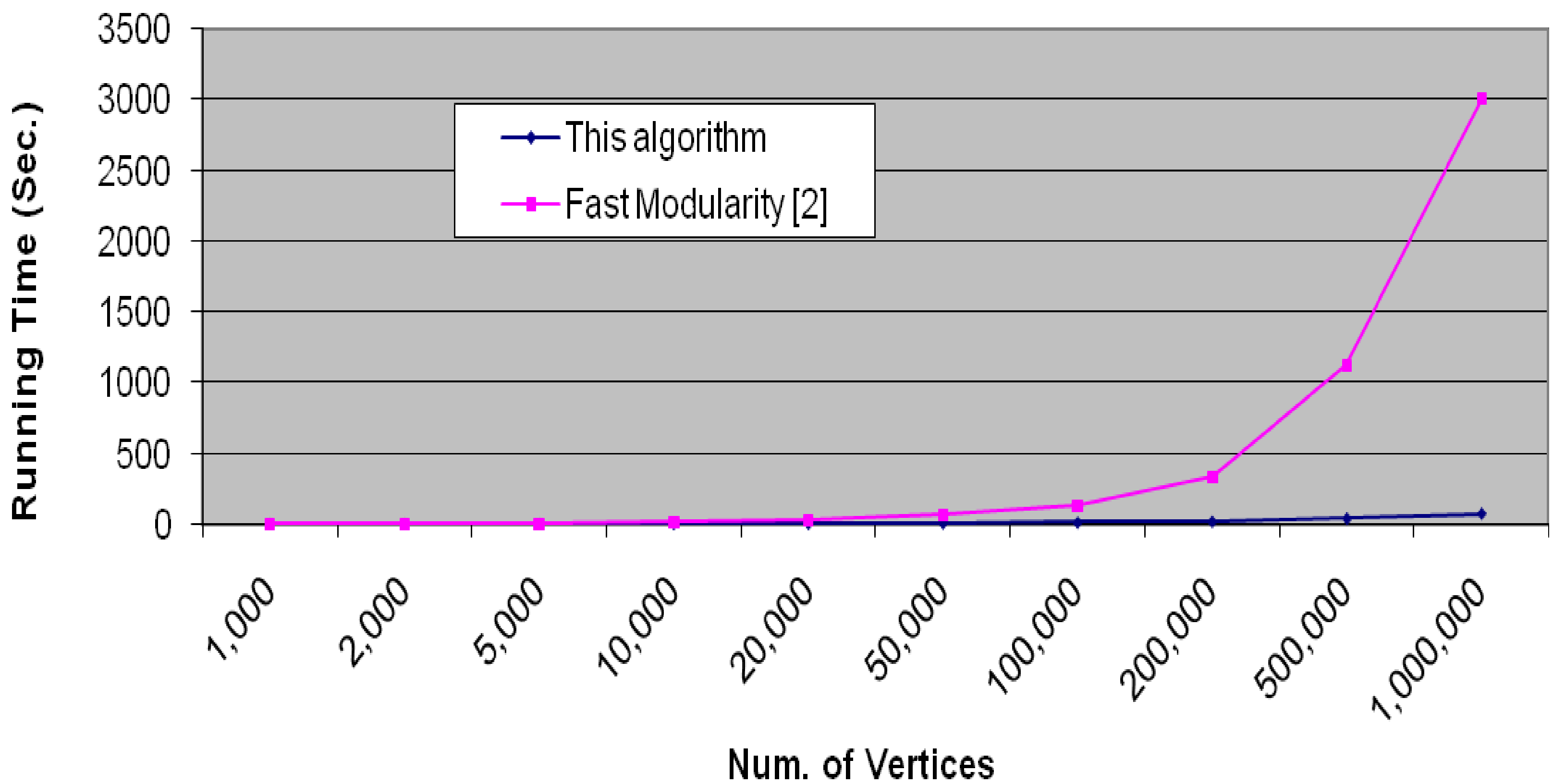
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



Running Time

- Running time = $O(|E|)$
- For sparse networks = $O(|V|)$



[2] A. Clauset, M. E. J. Newman, & C. Moore, *Phys. Rev. E* **70**, 066111 (2004).

Spectral Clustering

- Reference: ICDM'09 Tutorial by Chris Ding
- Example:
  - Clustering supreme court justices according to

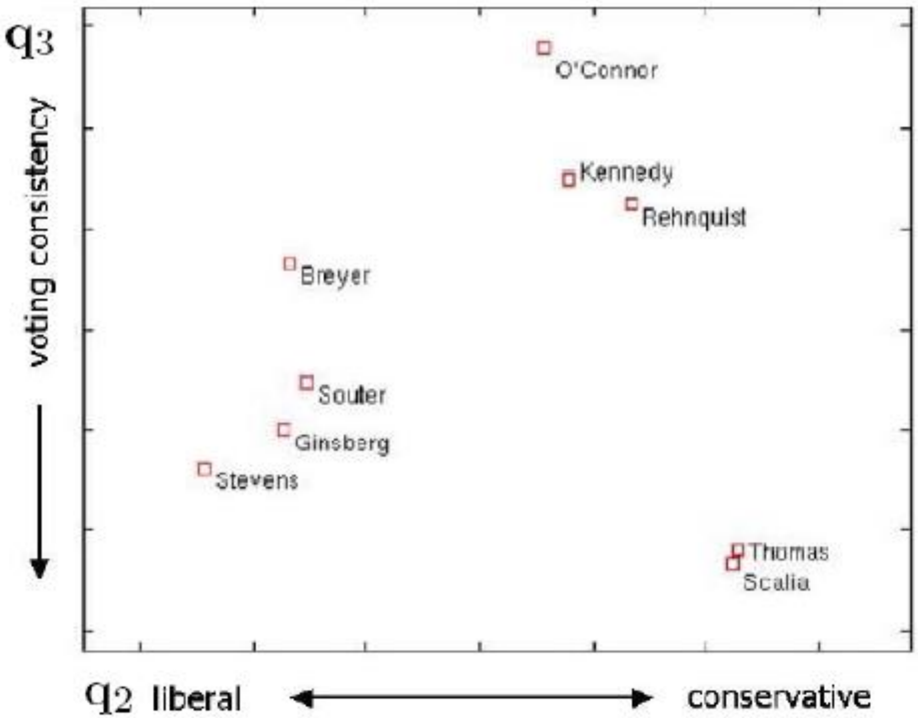
Number of times (%) two Justices voted in agreement

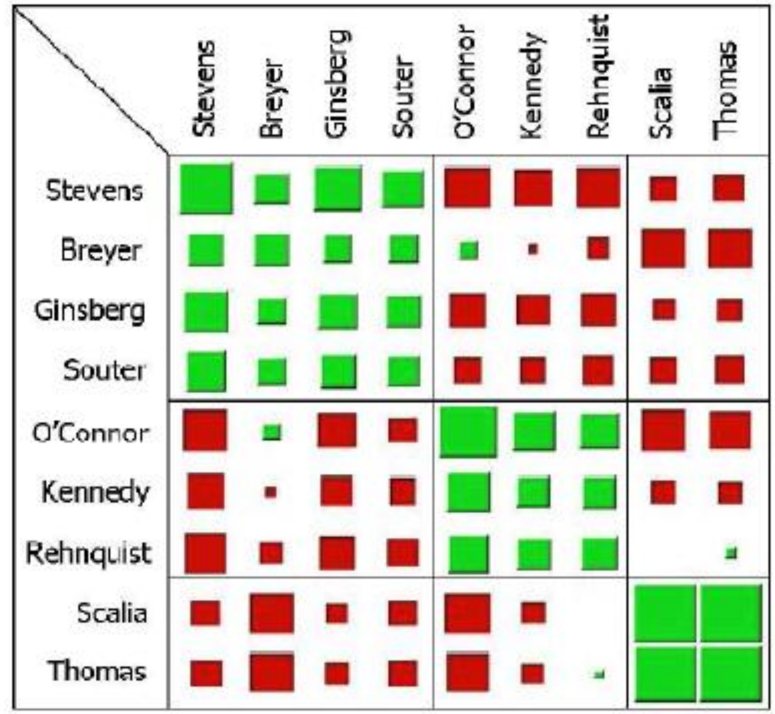
	Ste	Bre	Gin	Sou	O'Co	Ken	Reh	Sca	Tho
Stevens	–	62	66	63	33	36	25	14	15
Breyer	62	–	72	71	55	47	43	25	24
Ginsberg	66	72	–	78	47	49	43	28	26
Souter	63	71	78	–	55	50	44	31	29
O'Connor	33	55	47	55	–	67	71	54	54
Kennedy	36	47	49	50	67	–	77	58	59
Rehnquist	25	43	43	44	71	77	–	66	68
Scalia	14	25	28	31	54	58	66	–	79
Thomas	15	24	26	29	54	59	68	79	–

Table 1: From the voting record of Justices 1995 Term – 2004 Term, the number of times two justices voted in agreement (in percentage). (Data source: from July 2, 2005 *New York Times*. Originally from *Legal Affairs*; *Harvard Law Review*)

Example: Continue

$$C = \mathbf{q}_2\mathbf{q}_2^T + \mathbf{q}_3\mathbf{q}_3^T$$

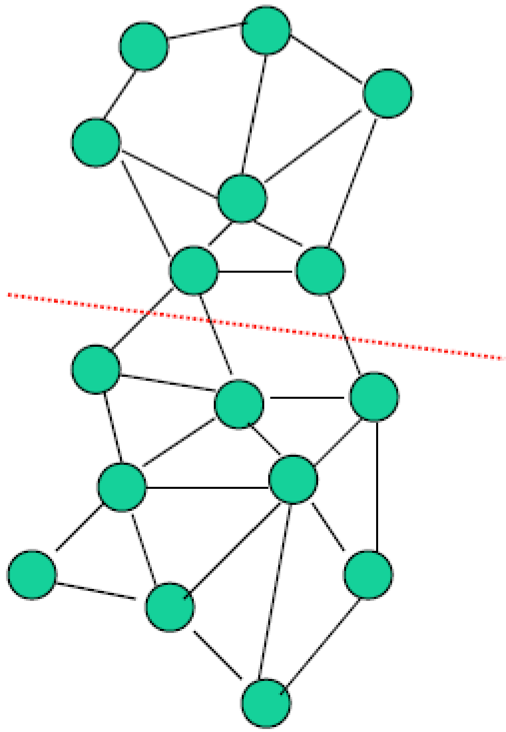




- Three groups in the Supreme Court:
  - Left leaning group, center-right group, right leaning group.

Spectral Graph Partition

- Min-Cut
  - Minimize the # of cut of edges



Objective Function

2-way Spectral Graph Partitioning

Partition membership indicator: $q_i = \begin{cases} 1 & \textbf{if } i \in A \\ -1 & \textbf{if } i \in B \end{cases}$

$$J = CutSize = \frac{1}{4}\sum_{i,j} w_{ij}[q_i - q_j]^2$$

$$= \frac{1}{4}\sum_{i,j} w_{ij}[q_i^2 + q_j^2 - 2q_i q_j] = \frac{1}{2}\sum_{i,j} q_i[d_i\delta_{ij} - w_{ij}]q_j$$

$$= \frac{1}{2} q^T (D - W) q$$

Relax indicators q_i from discrete values to continuous values, the solution for min $J(q)$ is given by the eigenvectors of

$$(D - W)q = \lambda q$$

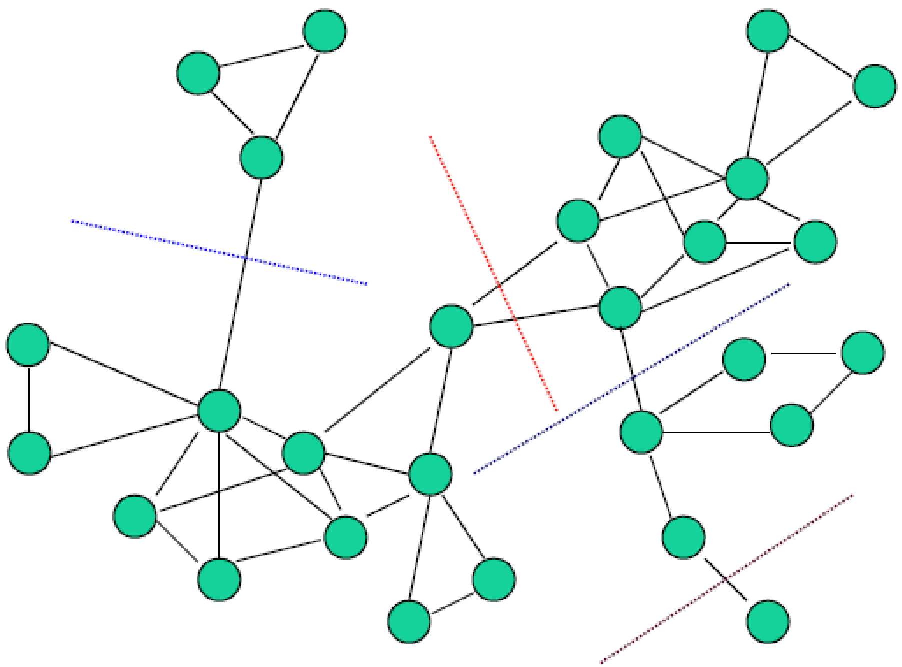
(Fiedler, 1973, 1975)

(Pothen, Simon, Liou, 1990)

Minimum Cut with Constraints

minimize cutsize without explicit size constraints

But where to cut ?



Need to balance sizes

New Objective Functions

$$s(A,B) = \sum_{i \in A} \sum_{j \in B} w_{ij}$$

- Ratio Cut (Hangen & Kahng, 1992)

$$J_{Rcut}(A,B) = \frac{s(A,B)}{|A|} + \frac{s(A,B)}{|B|}$$

- Normalized Cut (Shi & Malik, 2000)

$$d_A = \sum_{i \in A} d_i$$

$$J_{Ncut}(A,B) = \frac{s(A,B)}{d_A} + \frac{s(A,B)}{d_B}$$

$$= \frac{s(A,B)}{s(A,A)+s(A,B)} + \frac{s(A,B)}{s(B,B)+s(A,B)}$$

- Min-Max-Cut (Ding et al, 2001)

$$J_{MMC}(A,B) = \frac{s(A,B)}{s(A,A)} + \frac{s(A,B)}{s(B,B)}$$

Other References

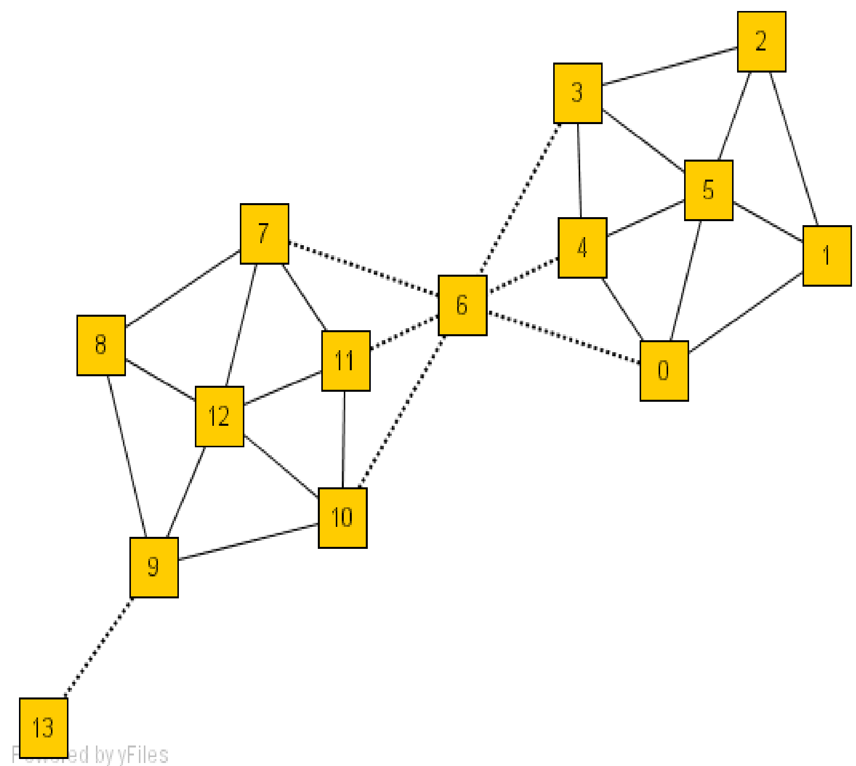
- A Tutorial on Spectral Clustering by U. Luxburg http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Luxburg07_tutorial_4488%5B0%5D.pdf

Mining Graph/Network Data: Part II

- Graph/Network Clustering
- Graph/Network Classification
- Summary

Label Propagation in the Network

- Given a network, some nodes are given labels, can we classify the unlabeled nodes by using link information?
  - E.g., Node 12 belongs to Class 1 and Node 5 Belongs to Class 2



Reference

- Learning from Labeled and Unlabeled Data with Label Propagation
  - By Xiaojin Zhu and Zoubin Ghahramani
  - http://www.cs.cmu.edu/~zhuxj/pub/CMU-CALD-02-107.pdf

Problem Formalization

- Given n nodes
  - l with labels ($Y_1, Y_2, \dots, Y_l$ *are known*)
  - u without labels ($Y_{l+1}, Y_{l+2}, \dots, Y_n$ are unknown)
  - Y *is the* $n \times C$ *label matrix*
    - C is the number of labels (classes)
- The adjacency matrix is W
- The probabilistic transition matrix T
  - $T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_k w_{kj}}$

The Label Propagation Algorithm

- Step 1: Propagate $Y \leftarrow TY$
- Step 2: Row-normalize Y
  - The summation of the probability of each object belonging to each class is 1
- Step 3: Reset the labels for the labeled nodes. Repeat 1-3 until Y converges

Mining Graph/Network Data: Part II

- Graph/Network Clustering
- Graph/Network Classification
- Summary

Summary

- Network Clustering
  - SCAN
  - Spectral clustering
- Network classification
  - Label propagation