CS145: INTRODUCTION TO DATA MINING

17: Graph and Network: Label Propagation and Spectral Clustering

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

November 29, 2021

Methods to Learn

	Vector Data	Set Data	Sequence Data/Time Series	Text Data	Graph Data
Classification	Logistic Regression; Decision Tree; NN			Naïve Bayes for Text	Label Propagation
Clustering	K-means; Mixture Models			PLSA	Spectral Clustering
Prediction	Linear Regression; Regression Tree; NN GLM*		AR Model		
Frequent Pattern Mining		Apriori; FP growth	GSP; PrefixSpan		
Similarity Search			DTW		P-PageRank
Ranking					PageRank

Graph Mining: Node Classification and Clustering

- Graph Laplacians
- Label Propagation
- Spectral Clustering
- Summary

What are Graph Laplacian Matrices?

- They are matrices defined as functions of graph adjacency or weight matrix

- A tool to study graphs

- There is a field called spectral graph theory studying those matrices

Examples of Graph Laplacians

- Given an undirected and weighted graph $G = (V, E)$, with weight matrix W
  - $W_{ij} = W_{ji} \geq 0$
  - n: total number of nodes
  - Degree for node $v_i \in V$: $d_i = \sum_j w_{ij}$
  - Degree matrix D: a diagonal matrix with degrees on the diagonal, i.e., $D_{ii} = d_i$ and $D_{ij} = 0\ if\ i \neq j$
- Three examples of graph Laplacians
  - The unnormalized graph Laplacian
    - $L = D - W$
  - The normalized graph Laplacians
    - Symmetric: $L_{sym} = D^{-1/2} L D^{-1/2}$
    - Random walk related: $L_{rw} = D^{-1} L$

The Unnormalized Graph Laplacian

- Definition: $L = D - W$
- Properties of L
  - For any vector $f \in R^n$

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

  - L is symmetric and positive semi-definite
  - The smallest eigenvalue of L is 0, and the corresponding eigenvector is the constant one vector $\mathbf{1}$
    - $\mathbf{1}$ is an all-one vector with n dimensions
    - An eigenvector can be scaled by multiplying a nonzero scalar α
  - L has n non-negative, real-valued eigenvalues:

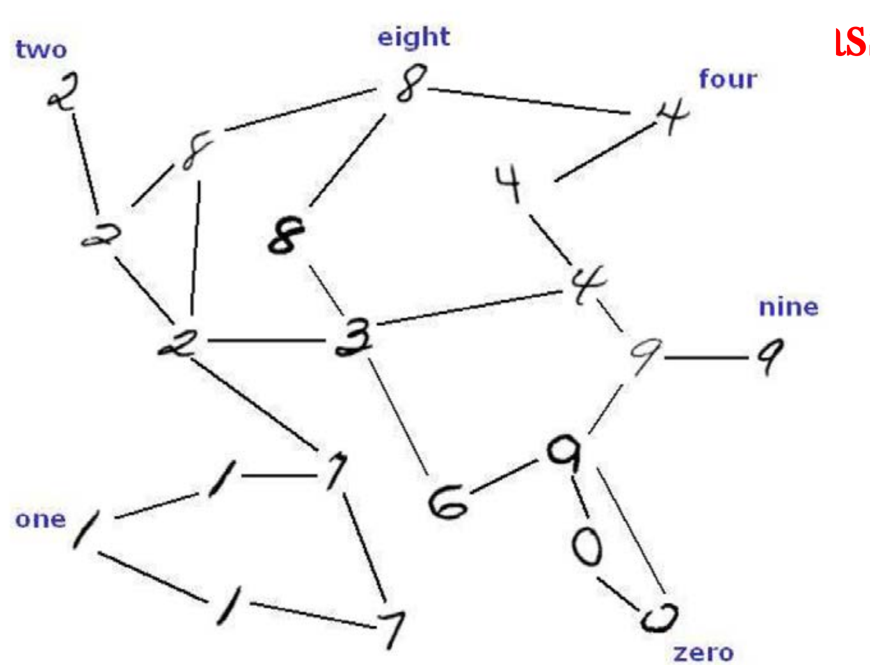
$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

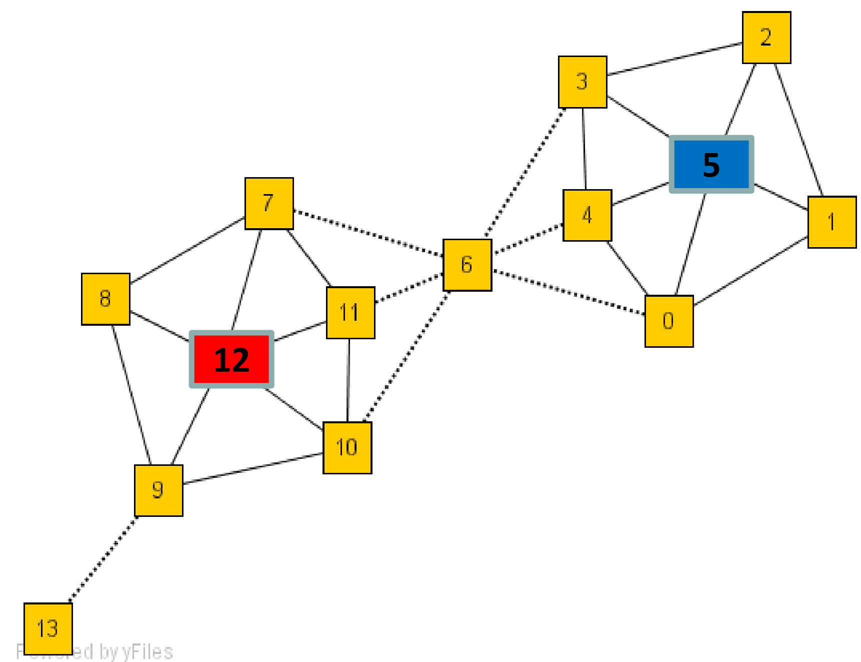
Graph Mining: Node Classification and Clustering

- Graph Laplacians
- Label Propagation
- Spectral Clustering
- Summary

Label Propagation in the Network

- Given an undirected network, some nodes are given labels, can we classify the unlabeled nodes by using link information?





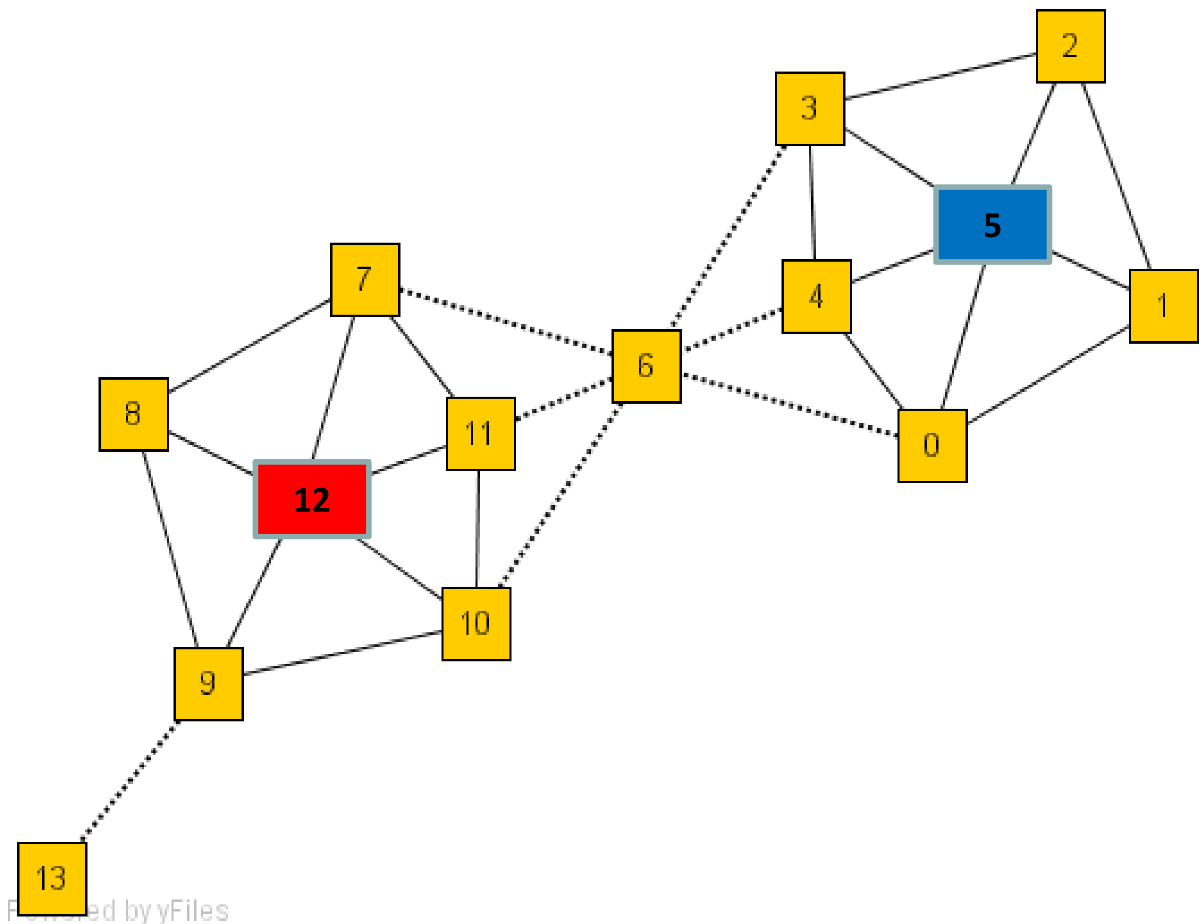
Problem Formalization for Label Propagation

- Given n nodes
  - l with labels (e.g., $Y_1, Y_2, \dots, Y_l$ *are known*)
  - u without labels (e.g., $Y_{l+1}, Y_{l+2}, \dots, Y_n$ are unknown)
  - Y *is the* $n \times K$ *label matrix*
    - K is the number of labels (classes)
    - Y_{ik} denotes the probability node i belonging to class k
- The weighted adjacency matrix is W
- The probabilistic transition matrix T
  - $T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{i'} w_{i'j}}$
  - Or $T = (D^{-1}W)^{\mathrm{T}}$

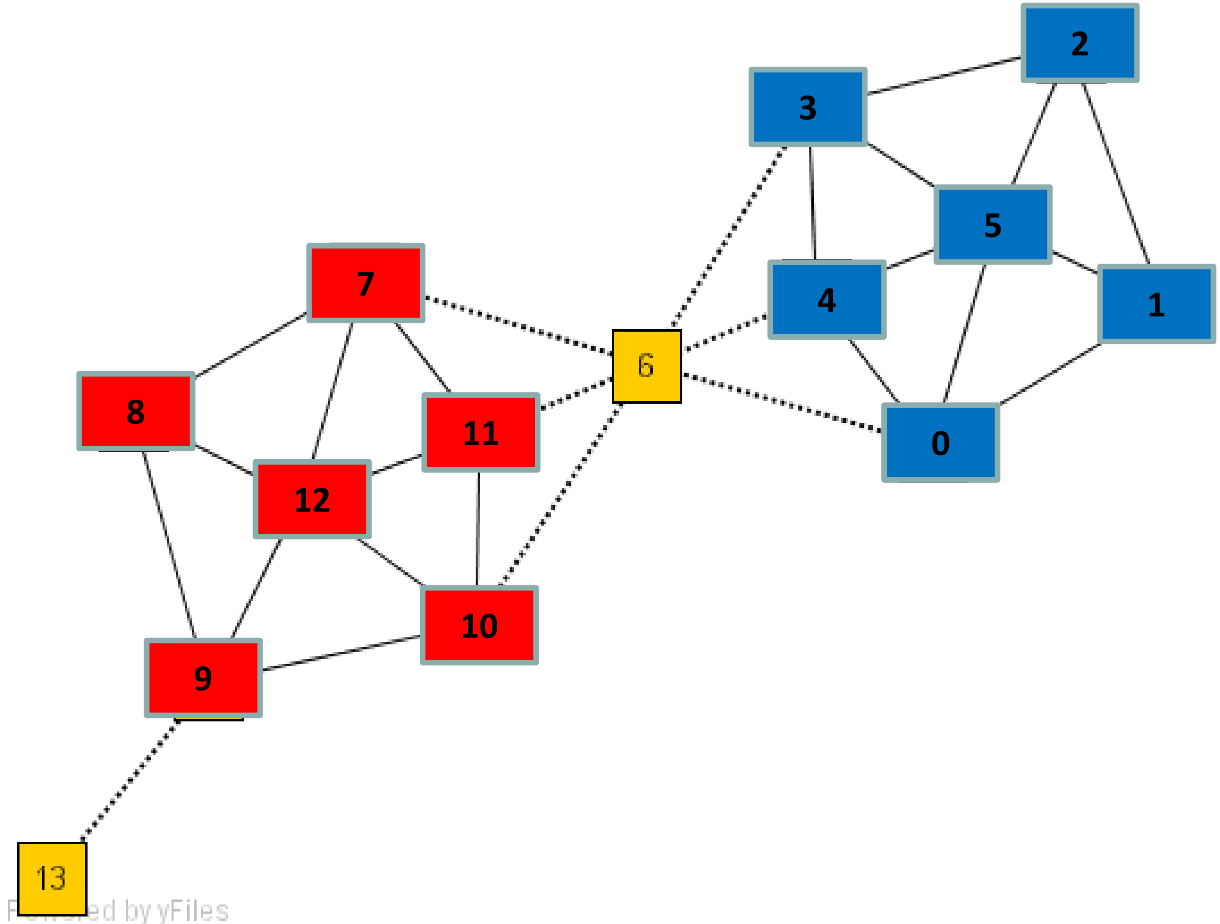
The Label Propagation Algorithm

- Step 1: Propagate $Y \leftarrow TY$
  - $Y_i = \sum_j T_{ij} Y_j = \sum_j P(j \to i) Y_j$
  - Initialization of Y for unlabeled ones is not important
- 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

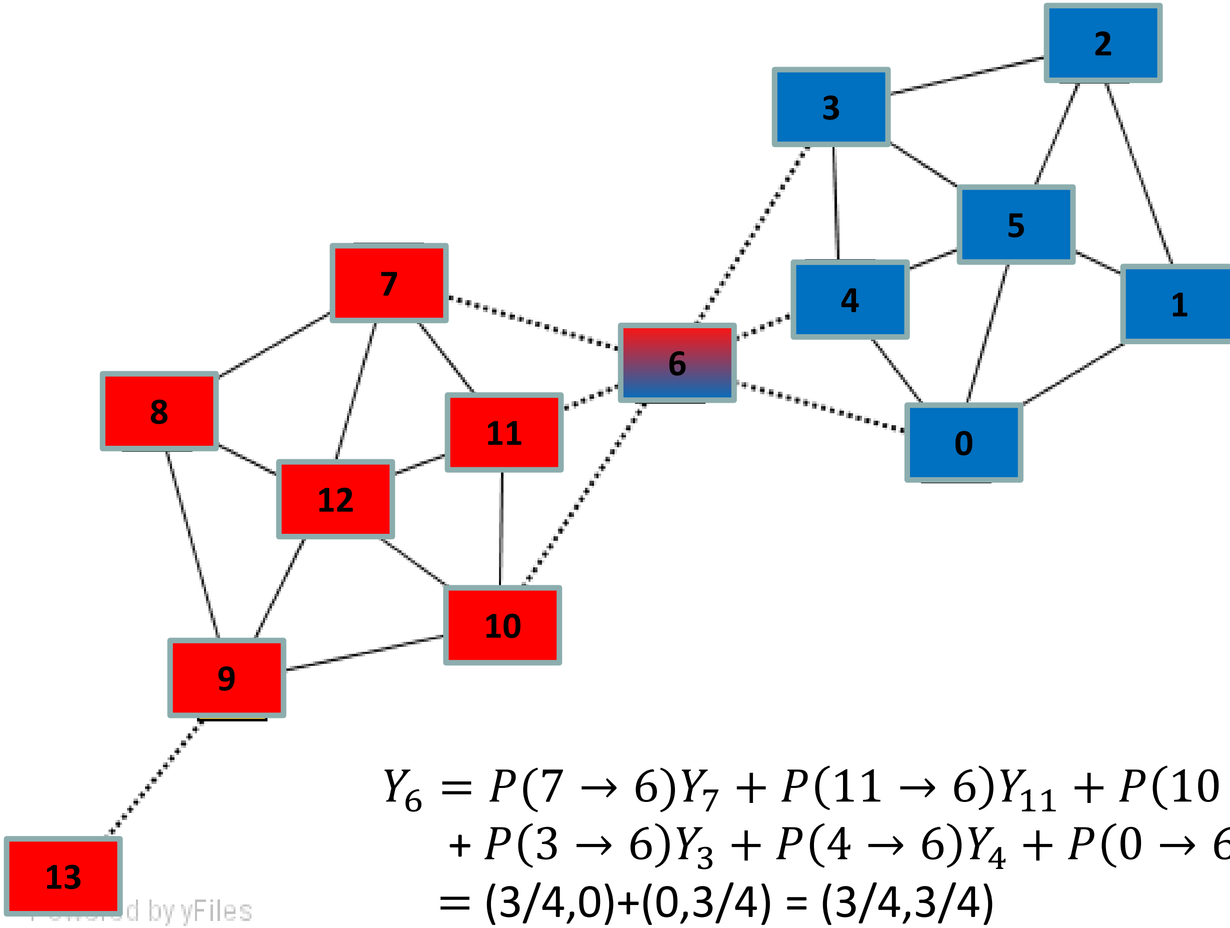
Example: Iter = 0



Example: Iter = 1



Example: Iter = 2



$$Y_6 = P(7 \to 6)Y_7 + P(11 \to 6)Y_{11} + P(10 \to 6)Y_{10}$$
$$+ P(3 \to 6)Y_3 + P(4 \to 6)Y_4 + P(0 \to 6)Y_0$$
$$= (3/4,0)+(0,3/4) = (3/4,3/4)$$

After normalization, $Y_6 = (\frac{1}{2}, \frac{1}{2})$

*Other Label Propagation Algorithms

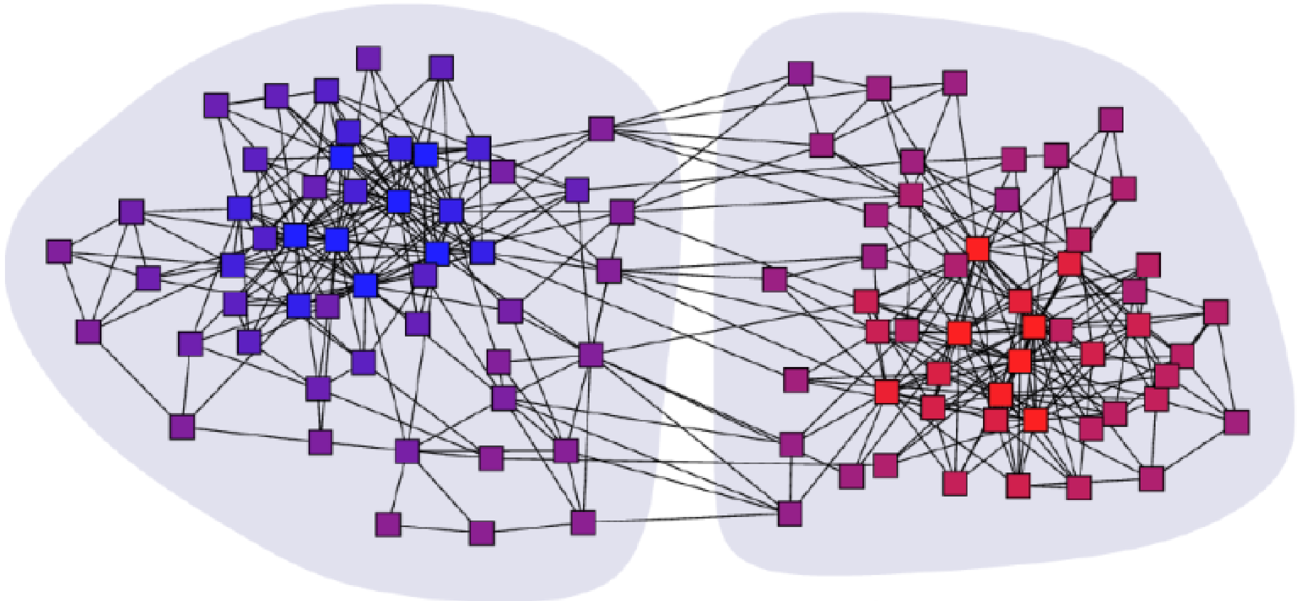
- Energy minimizing and harmonic function
  - Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions
    - By Xiaojin Zhu et al., ICML'03
    - https://www.aaai.org/Papers/ICML/2003/ICML03-118.pdf
- Graph regularization
  - Learning with Local and Global Consistency
    - By Denny Zhou et al., NIPS'03
    - http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf

Graph Mining: Node Classification and Clustering

- Graph Laplacians
- Label Propagation
- Spectral Clustering
- 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]

Example of Graph Clustering

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

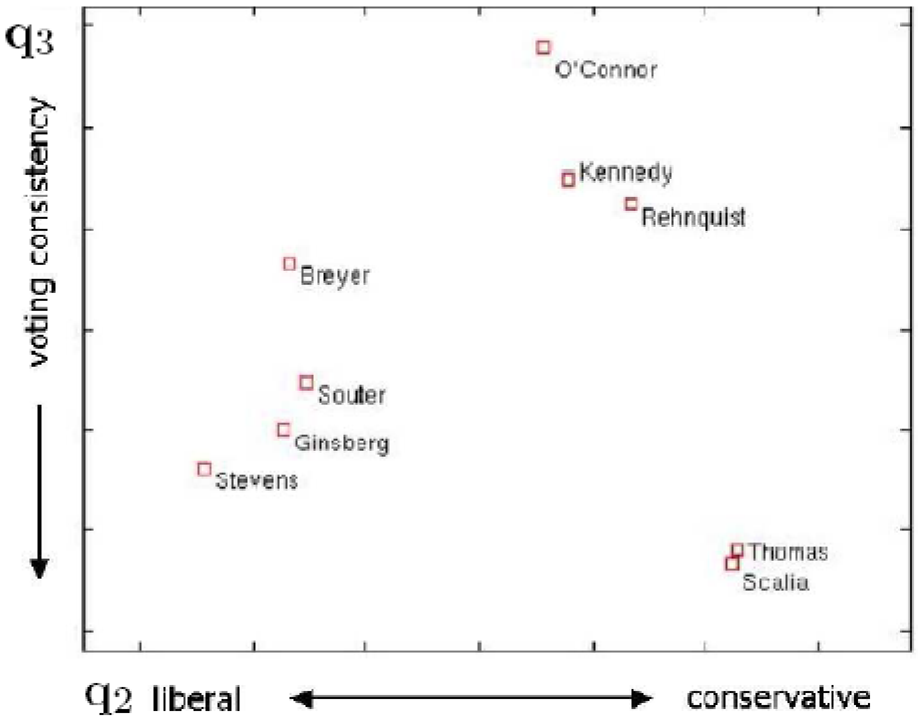
W =

Number of times (%) two Justices voted in agreement

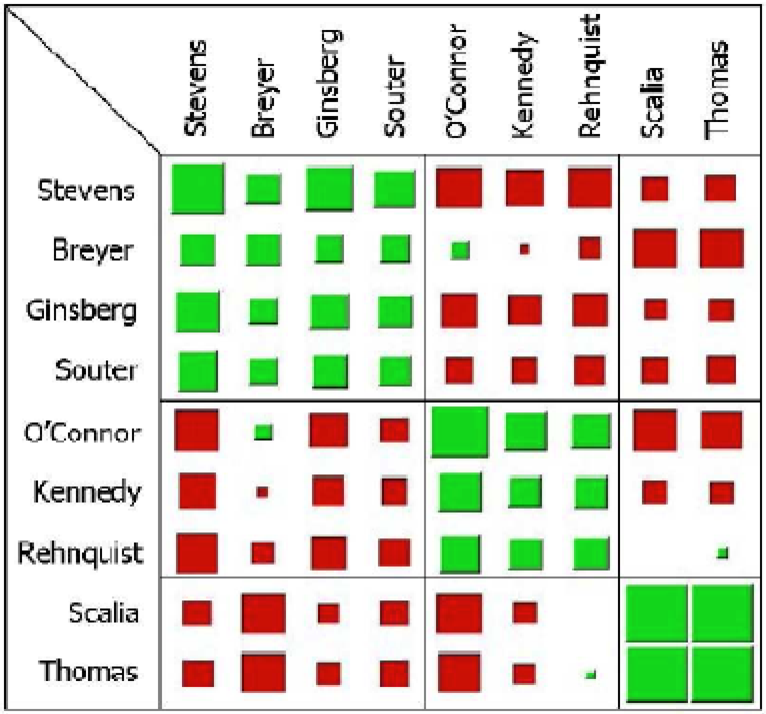
	Ste	Bre	Gin	Sou	O'Co	Ken	Reh	Sca	Tho
Stevens	–	62	66	63	33	36	25	14	15
Breyer	62	–	72	71	55	47	43	25	24
Ginsberg	66	72	–	78	47	49	43	28	26
Souter	63	71	78	–	55	50	44	31	29
O'Connor	33	55	47	55	–	67	71	54	54
Kennedy	36	47	49	50	67	–	77	58	59
Rehnquist	25	43	43	44	71	77	–	66	68
Scalia	14	25	28	31	54	58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

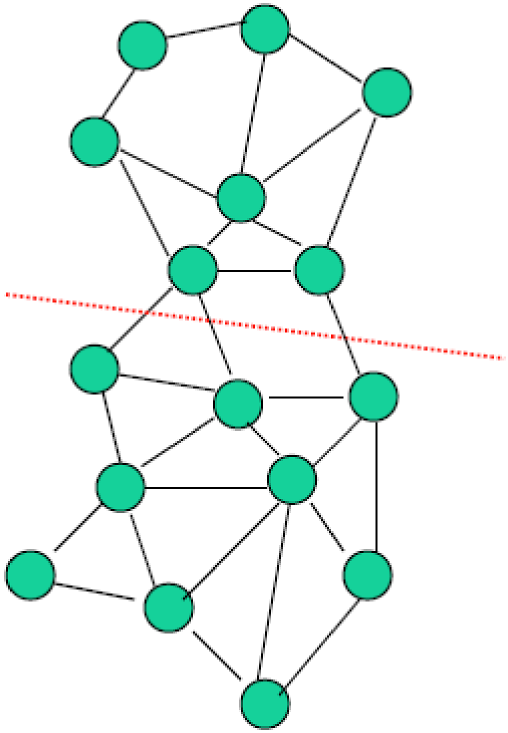
- Goal: cluster nodes in the graph with weight matrix W into k clusters
- Idea: Leverage the first k eigenvectors of $L = D - W$
- Major steps:
  - Compute the first k eigenvectors, $\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_k$, of L
  - Each node i is then represented by k values $y_i = (q_{1i}, q_{2i}, \dots, q_{ki})$
  - Cluster y_i's using k-means

*Variants of Spectral Clustering Algorithms

- Normalized spectral clustering according to Shi and Malik (2000)
  - Compute the first k eigenvectors, $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$, of L_{rw}
- Normalized spectral clustering according to Ng, Jordan, and Weiss (2002)
  - Compute the first k eigenvectors, $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$, of L_{sym}
  - Normalizing y_i to have norm 1

Connections to Graph Cuts

- Min-Cut
  - Minimize the # of cut of edges to partition a graph into 2 disconnected components



Objective Function of Min-Cut

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

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

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

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

- The optimization problem: $$\min J(q) \quad s.t.\, q^T q = n$$

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)

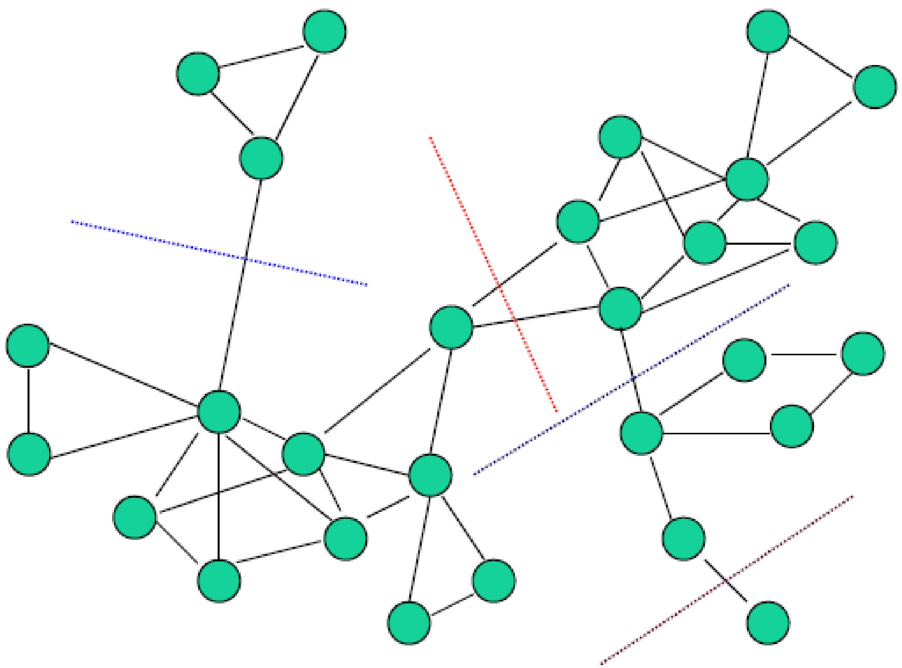
Algorithm

- Step 1:
  - Calculate Graph Laplacian matrix: $L = D - W$
- Step 2:
  - Calculate the second eigenvector q
    - Q: Why second?
- Step 3:
  - Bisect q (e.g., 0) to get two clusters

*Minimum Cut with Constraints

minimize cutsize without explicit size constraints

But where to cut ?



Need to balance sizes

*Other Objective Functions

- Ratio Cut (Hangen & Kahng, 1992)

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

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

- 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)

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

Graph Mining: Node Classification and Clustering

- Graph Laplacians
- Label Propagation
- Spectral Clustering
- Summary

Summary

- Graph Laplacians are keys to understand graphs
  - Unnormalized graph Laplacians
  - Normalized graph Laplacians
- Label propagation
  - Semi-supervised node classification on graphs
  - Also related to graph Laplacians
- Spectral clustering
  - Leverage eigenvectors of graph Laplacians to conduct clustering on graphs
  - Has close relationship with graph cuts

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
- 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
- Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions
  - By Xiaojin Zhu et al., ICML'03
  - https://www.aaai.org/Papers/ICML/2003/ICML03-118.pdf
- Learning with Local and Global Consistency
  - By Denny Zhou et al., NIPS'03
  - http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf