

CS247: ADVANCED DATA MINING

Graph and Network: Spectral Clustering and Label Propagation

Instructor: Yizhou Sun


yzsun@ccs.neu.edu

May 10, 2021

Methods to Learn

	Vector Data	Text Data	Graph & Network	Recommender Systems
Classification	Naïve Bayes; Logistic Regression; NN		Label Propagation	
Clustering	K-means; Mixture Models	PLSA; LDA	Spectral Clustering	Matrix Factorization
Prediction	NN			Collaborative Filtering; Factorization machine; Hybrid CF; Recommendation with graph regularization
Ranking			PageRank	
Similarity Search			P-PageRank	
Representation Learning		Word embedding	Network embedding	Deep collaborative learning

Graph: Clustering and Classification

- Graph Laplacians 
- Spectral Clustering
- Label Propagation
- 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 $f \in R^n$

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

- L is symmetric and positive semi-definite
- The smallest eigenvalue of L is 0, and the corresponding eigenvector is the constant one vector **1**
 - **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:

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

Question

- Will self loops in graph change L ?

Number of Connected Components

- The multiplicity k of the eigenvalue 0 of L equals the number of connected components
 - Consider $k = 1$, i.e., a connected graph, and f , the corresponding eigenvector for 0

$$0 = f' L f = \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

- Every element of f has to be equal to each other

K connected components

- The graph can be represented as a block diagonal matrix, and so do matrix L

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

- For each block L_i , it has eigenvalue 0 with corresponding constant one eigenvector
- For L , it has k eigenvalues with 0, with corresponding eigenvectors as indicator vectors
 - $\mathbf{1}_{A_i}$ is an indicator vector, with ones for nodes in A_i , i.e., the nodes in component i, and zeros elsewhere

The normalized graph Laplacians

- Symmetric:

- $L_{sym} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$

- Random walk related:

- $L_{rw} = D^{-1}L = I - D^{-1}W$

Properties of L_{sym} and L_{rw}

- For any vector $f \in R^n$

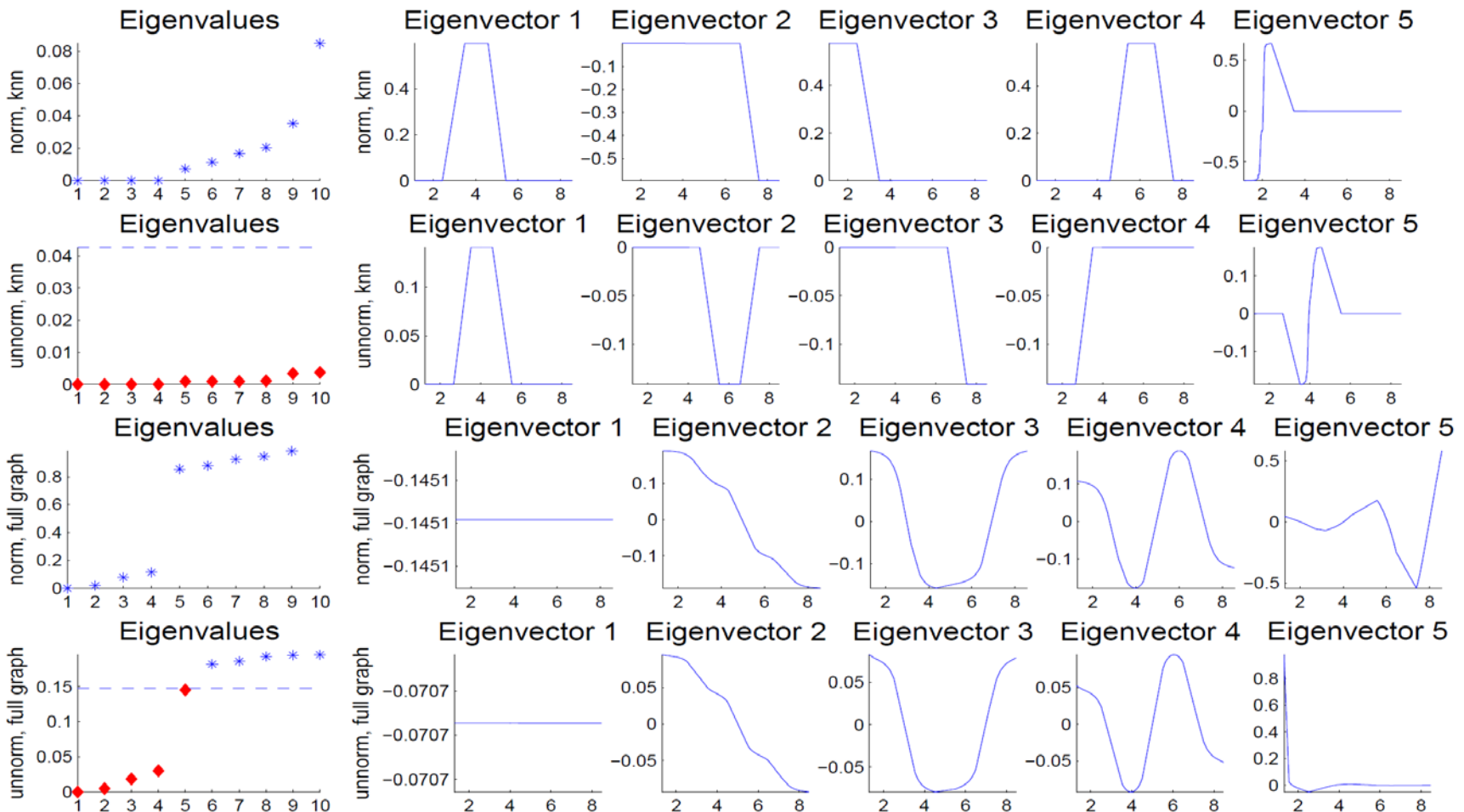
$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

- λ is an eigenvalue of L_{rw} with eigenvector $v \Leftrightarrow \lambda$ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2}v$
- λ is an eigenvalue of L_{rw} with eigenvector $v \Leftrightarrow \lambda$ and v solves the generalized eigen problem $Lv = \lambda Dv$
- 0 is an eigenvalue of L_{rw} with eigenvector $\mathbf{1}$. 0 is an eigenvalue of L_{sym} with eigenvector $D^{1/2}\mathbf{1}$
- L_{sym} and L_{rw} are positive semi-definite, and have n non-negative real-valued eigenvalues


$$0 = \lambda_1 \leq \dots \leq \lambda_n$$

Example

- Graph built upon Gaussian mixture model with four components

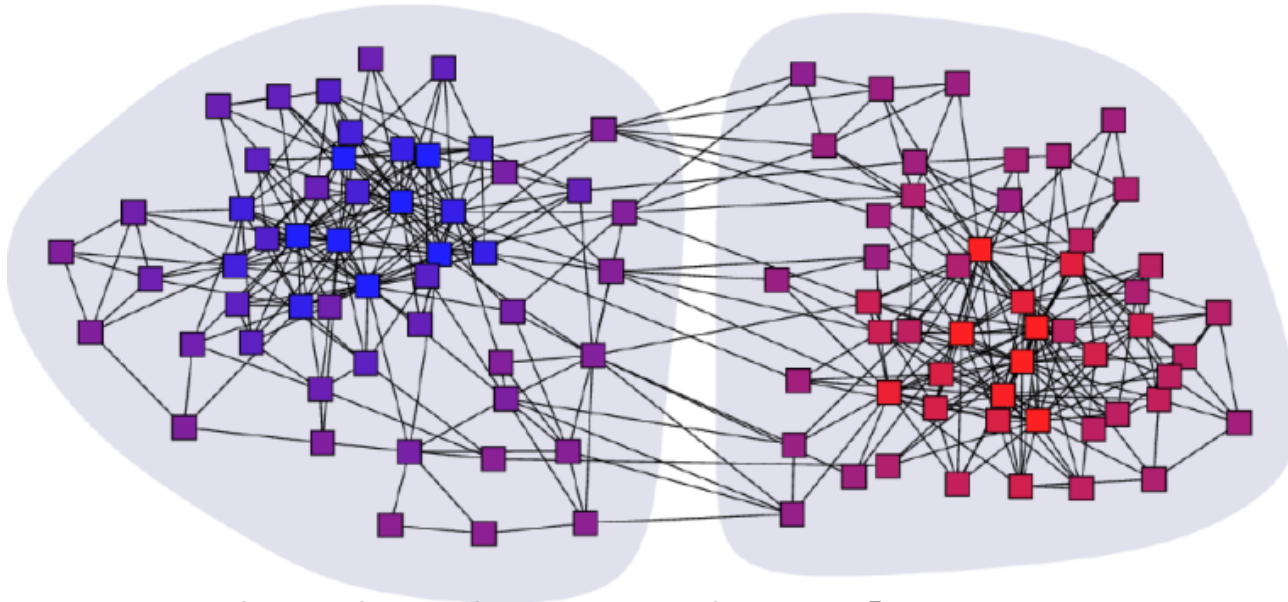


Graph: Clustering and Classification

- Graph Laplacians
- Spectral Clustering 
- Label Propagation
- 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

Number of times (%) two Justices voted in agreement

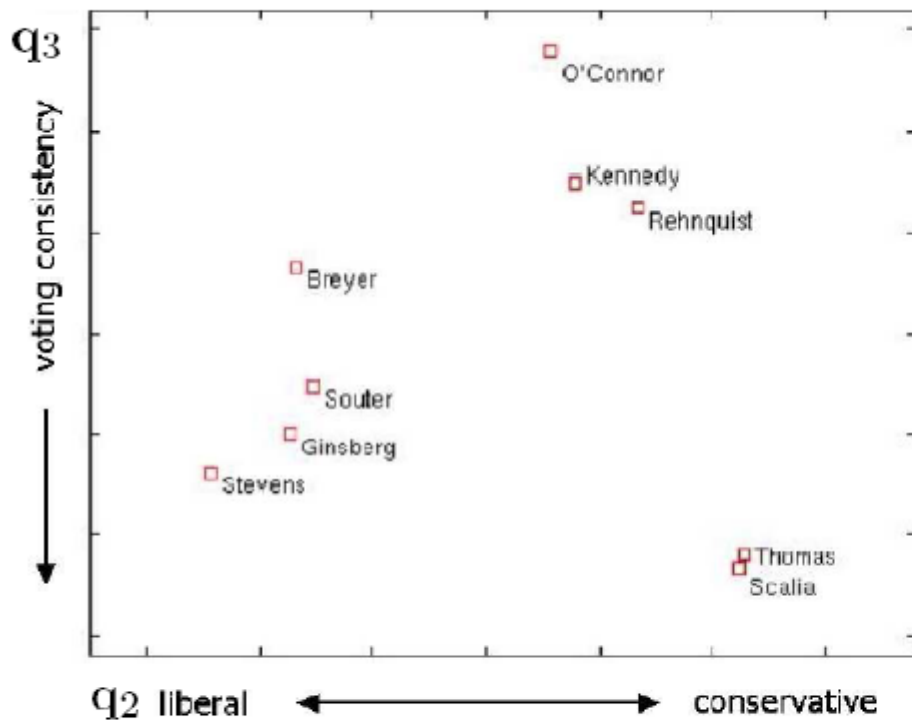
	Ste	Bre	Gin	Sou	O'Co	Ken	Reh	Scal	Tho
Stevens	–	62	66	63	33	36	25	14	15
Breyer	62	–	72	71	55	47	43	25	24
Ginsberg	66	72	–	78	47	49	43	28	26
Souter	63	71	78	–	55	50	44	31	29
O'Connor	33	55	47	55	–	67	71	54	54
Kennedy	36	47	49	50	67	–	77	58	59
Rehnquist	25	43	43	44	71	77	–	66	68
Scalia	14	25	28	31	54	58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

$W =$

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

$$C = q_2 q_2^T + q_3 q_3^T$$



	Stevens	Breyer	Ginsberg	Souter	O'Connor	Kennedy	Rehnquist	Scalia	Thomas
Stevens	Green	Green	Green	Green	Red	Red	Red	Red	Red
Breyer	Green	Green	Green	Green	Green	Red	Red	Red	Red
Ginsberg	Green	Green	Green	Green	Red	Red	Red	Red	Red
Souter	Green	Green	Green	Green	Red	Red	Red	Red	Red
O'Connor	Red	Green	Red	Red	Green	Green	Green	Red	Red
Kennedy	Red	Red	Red	Red	Green	Green	Green	Red	Red
Rehnquist	Red	Red	Red	Red	Green	Green	Green	Red	Red
Scalia	Red	Red	Red	Red	Red	Red	Red	Green	Green
Thomas	Red	Red	Red	Red	Red	Red	Red	Green	Green

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

Spectral Clustering Algorithms

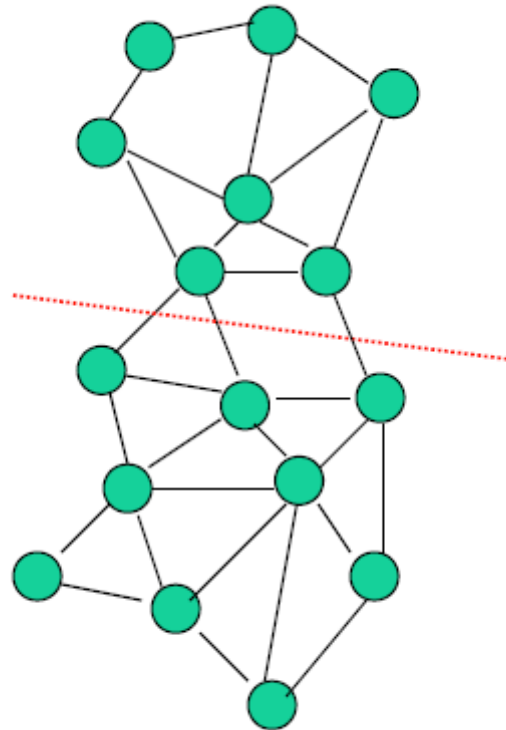
- Goal: cluster nodes in the graph into k clusters
- Idea: Leverage the first k eigenvectors of L
- Major steps:
 - Compute the first k eigenvectors, v_1, v_2, \dots, v_k , of L
 - Each node i is then represented by k values $y_i = (v_{1i}, v_{2i}, \dots, v_{ki})$
 - Cluster y_i 's using k-means

Variants of Spectral Clustering Algorithms

- Normalized spectral clustering according to Shi and Malik (2000)
 - Compute the first k eigenvectors, v_1, v_2, \dots, v_k , of L_{rw}
- Normalized spectral clustering according to Ng, Jordan, and Weiss (2002)
 - Compute the first k eigenvectors, v_1, v_2, \dots, v_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: $q_i = \begin{cases} 1 & \text{if } i \in A \\ -1 & \text{if } i \in B \end{cases}$

$$\begin{aligned} J = \text{CutSize} &= \frac{1}{4} \sum_{i,j} w_{ij} [q_i - q_j]^2 \\ &= \frac{1}{4} \sum_{i,j} w_{ij} [q_i^2 + q_j^2 - 2q_i q_j] = \frac{1}{2} \sum_{i,j} q_i [d_i \delta_{ij} - w_{ij}] q_j \\ &= \frac{1}{2} q^T (D - W) q \end{aligned}$$

- The optimization problem:
$$\begin{aligned} &\min J(q) \\ &s. t. q^T q = n \end{aligned}$$

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)

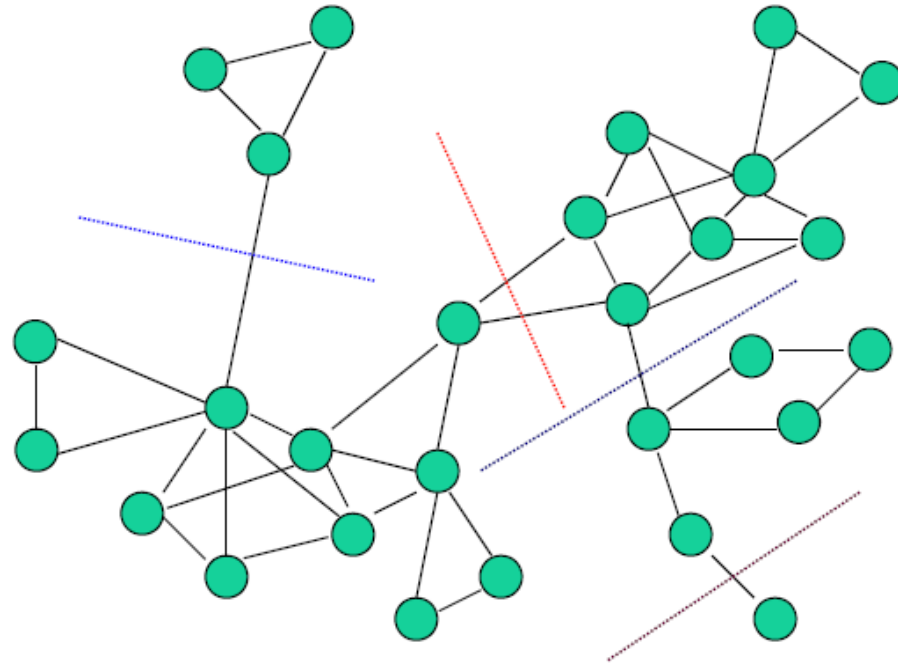
Algorithm

- 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 cutsizes 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)

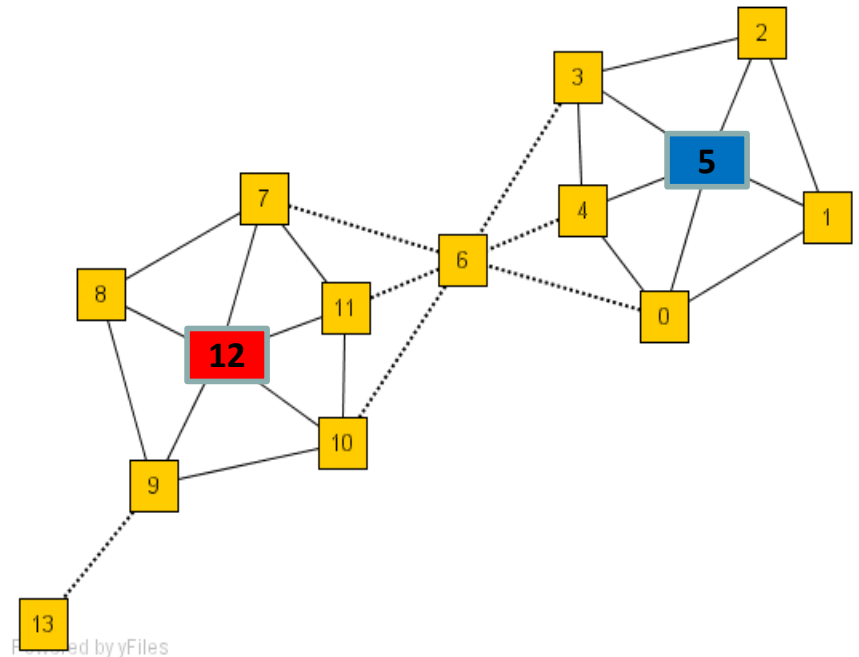
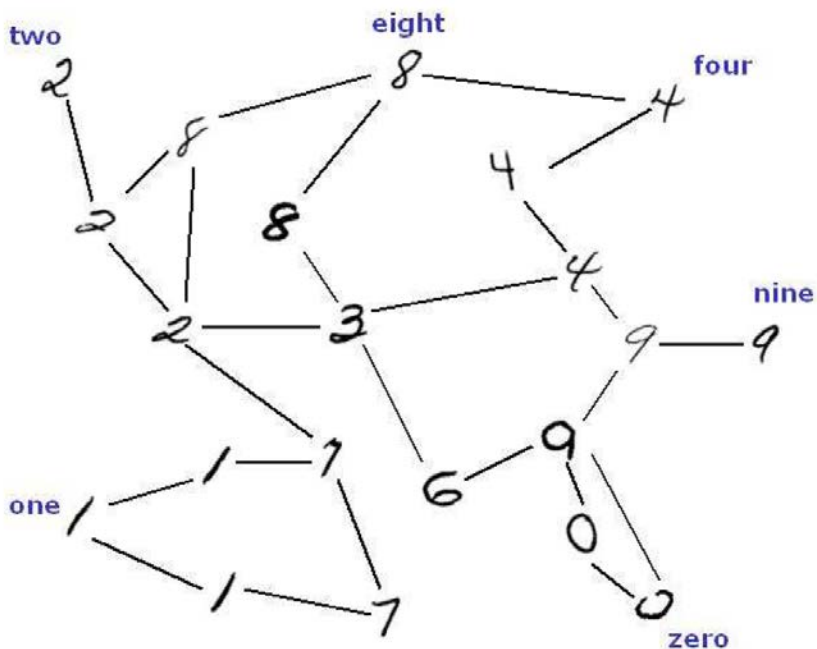
$$J_{MMC}(A,B) = \frac{s(A,B)}{s(A,A)} + \frac{s(A,B)}{s(B,B)}$$

Graph: Clustering and Classification

- Graph Laplacians
- Spectral Clustering
- Label Propagation 
- 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**, Node 5 Belongs to **Class 2**



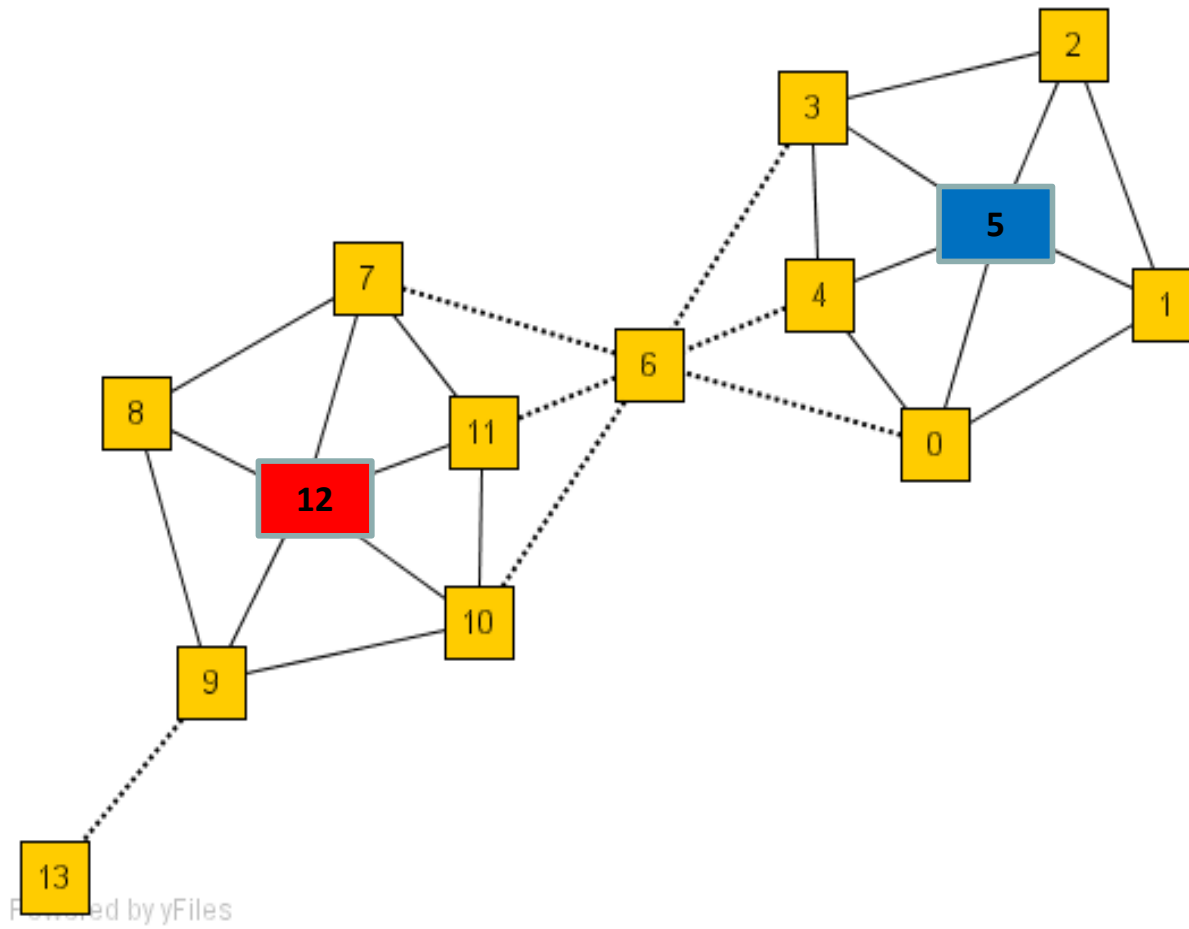
Problem Formalization for Label Propagation

- 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
 - $T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{i'} w_{i'j}}$

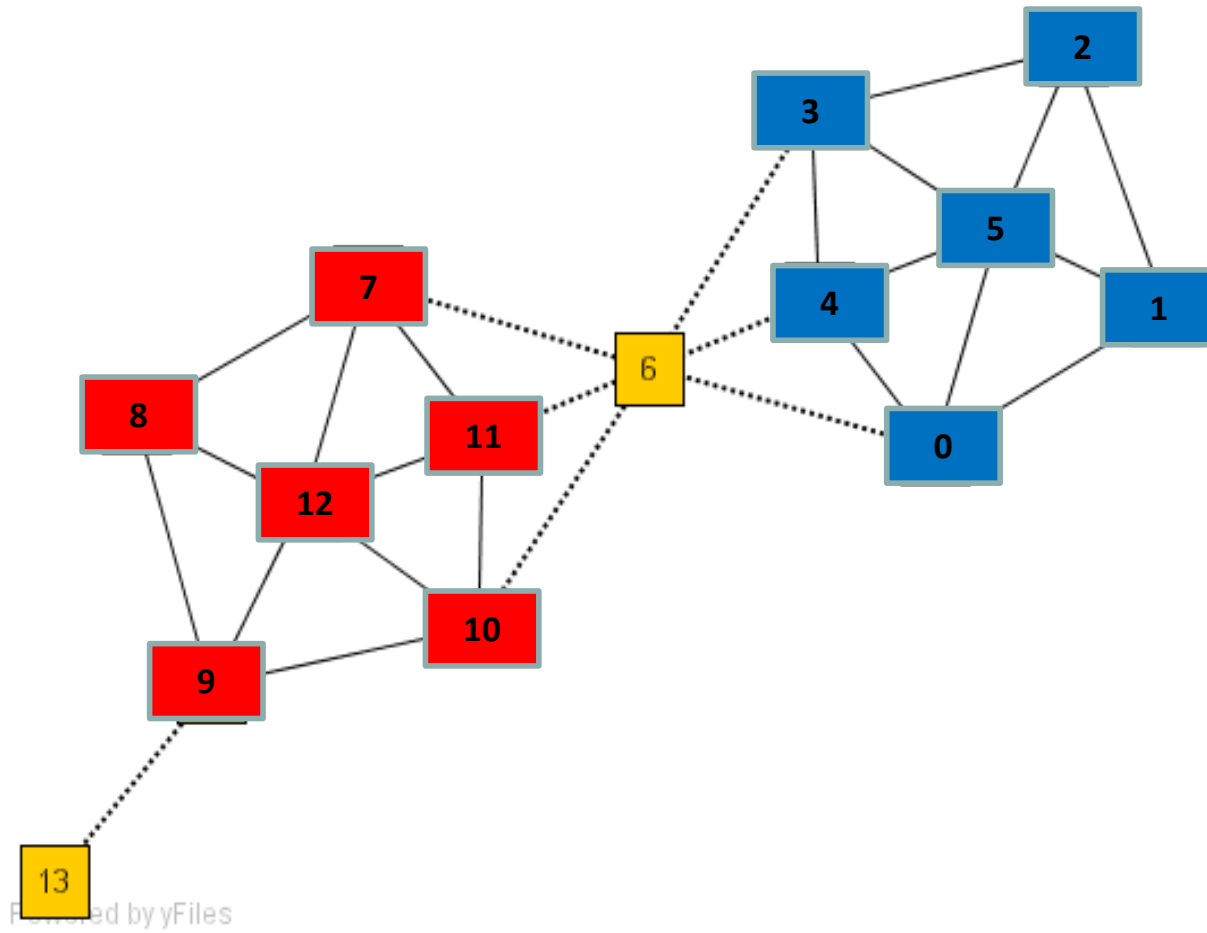
The Label Propagation Algorithm

- Step 1: Propagate $Y \leftarrow TY$
 - $Y_i = \sum_j T_{ij} Y_j = \sum_j P(j \rightarrow i) Y_j$
 - 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

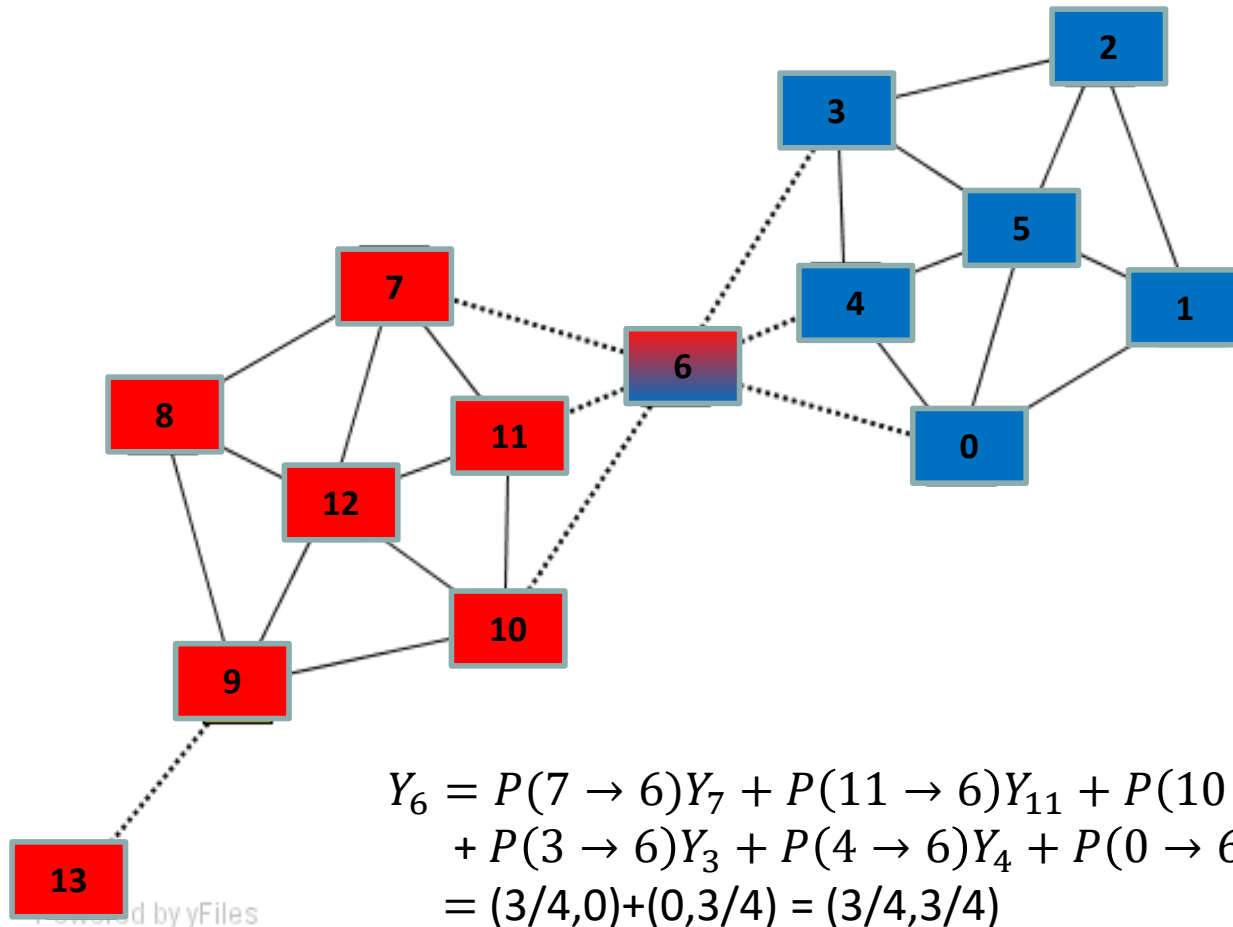
Example: Iter = 0



Example: Iter = 1



Example: Iter = 2



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

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>

From Energy Minimizing Perspective

- Consider a binary classification problem
 - $y \in \{0,1\}$
- Let $f: V \rightarrow R$, which maps a node to a real number
 - $f(i) = y_i$, if i is labeled
 - $f = \begin{pmatrix} f_l \\ f_u \end{pmatrix}$
- The energy function
 - $$E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2$$
 - Intuition: if two nodes are connected, they should share similar labels

Minimizing Energy Function Results in Harmonic Function

- Note $E(f) = f'(D - W)f = f'Lf!$
- *Goal: find f such that $E(f)$ is minimized, and f_l is fixed*
- *Solution:*
 - $(D - W)f = 0$

Solve f_u

- Consider W as a block matrix

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}$$

- Closed form solution of f_u :

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l = (I - P_{uu})^{-1} P_{ul} f_l$$

- Where $P = D^{-1}W$

- Iterative solution of f_u :

- $f_u = (\sum_{t=0} P_{uu}^t) P_{ul} f_l \Rightarrow f_u^t = P_{uu} f_u^{t-1} + P_{ul} f_l$

- As $(I - P_{uu})^{-1} = I + P_{uu} + P_{uu}^2 + \dots$

From Graph Regularization Perspective


- Let F be $n \times K$ matrix with nonnegative entries
 - For node i , pick the label k such that F_{ik} is the maximum among all the k
- Let Y be $n \times K$ matrix with $y_{ik} = 1$, if node i is labeled as class k , and 0 otherwise.
- Cost function:
- **Smoothness constraint + fitting constraint**

$$Q(F) = \frac{1}{2} \left(\sum_{i,j=1}^n W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^n \|F_i - Y_i\|^2 \right)$$

Solve F

- Note the first component is related to $\text{trace}(F' L_{sym} F)$
- Closed form solution:
 - $\frac{\partial Q}{\partial F} \Big|_{F=F^*} = F^* - SF^* + \mu(F^* - Y) = 0$
 - $\Rightarrow F^* = (1 - \alpha)(I - \alpha S)^{-1} Y$
 - Where $S = D^{-1/2} W D^{-1/2}$, $\alpha = \frac{1}{1+\mu}$ is a value between 0 and 1
- Iterative solution:
 - $F(t+1) = \alpha S F(t) + (1 - \alpha) Y$

Graph: Clustering and Classification

- Graph Laplacians
- Spectral Clustering
- Label Propagation
- Summary 

Summary

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

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>