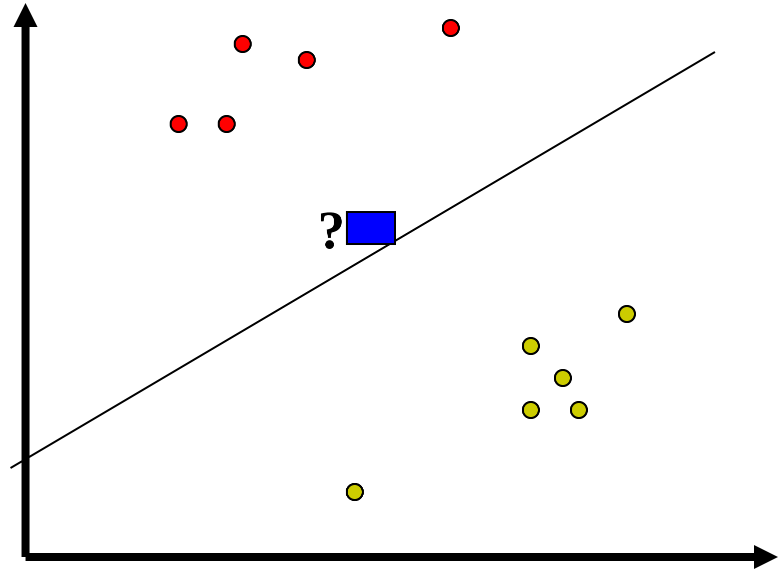
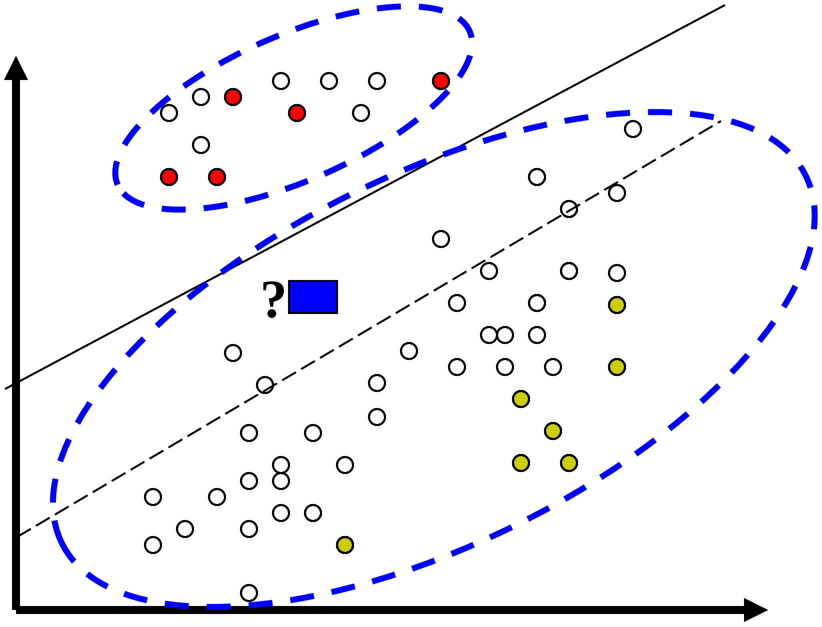
Graph-based Semi-Supervised Learning: Label Propagation

Clustering Assumption



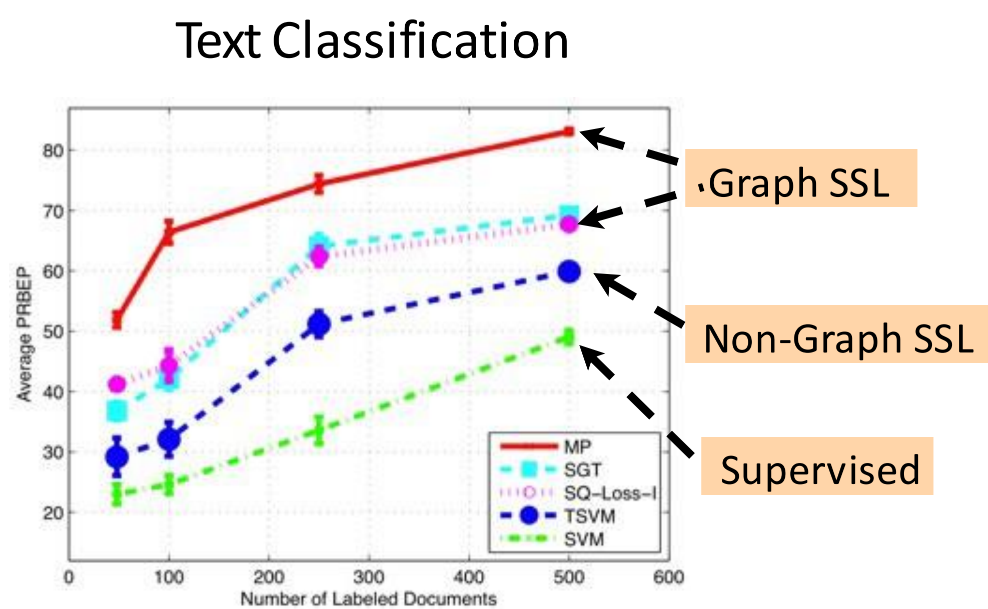
Clustering Assumption



- Clusters are separated through low-density regions

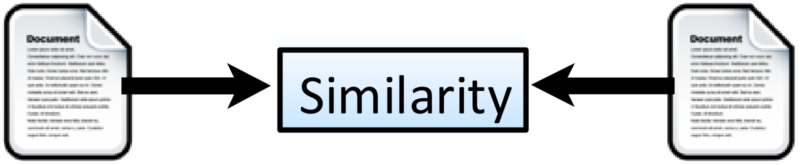
Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Effective in practice

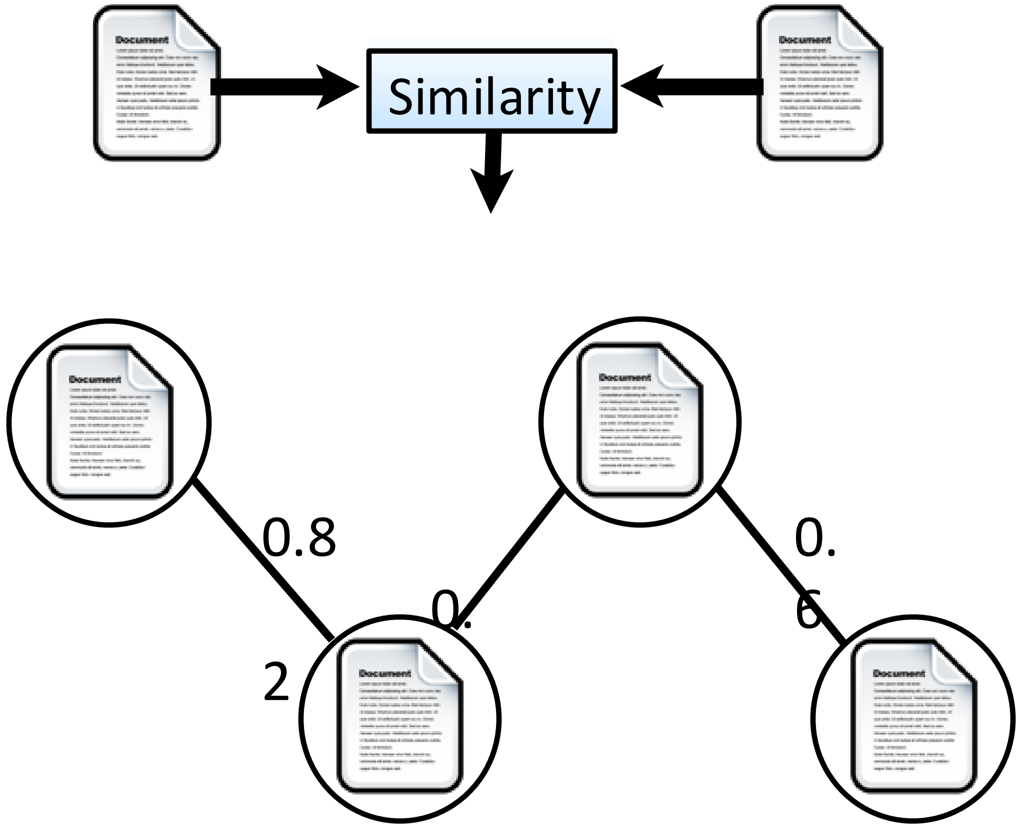


Graph-based SSL

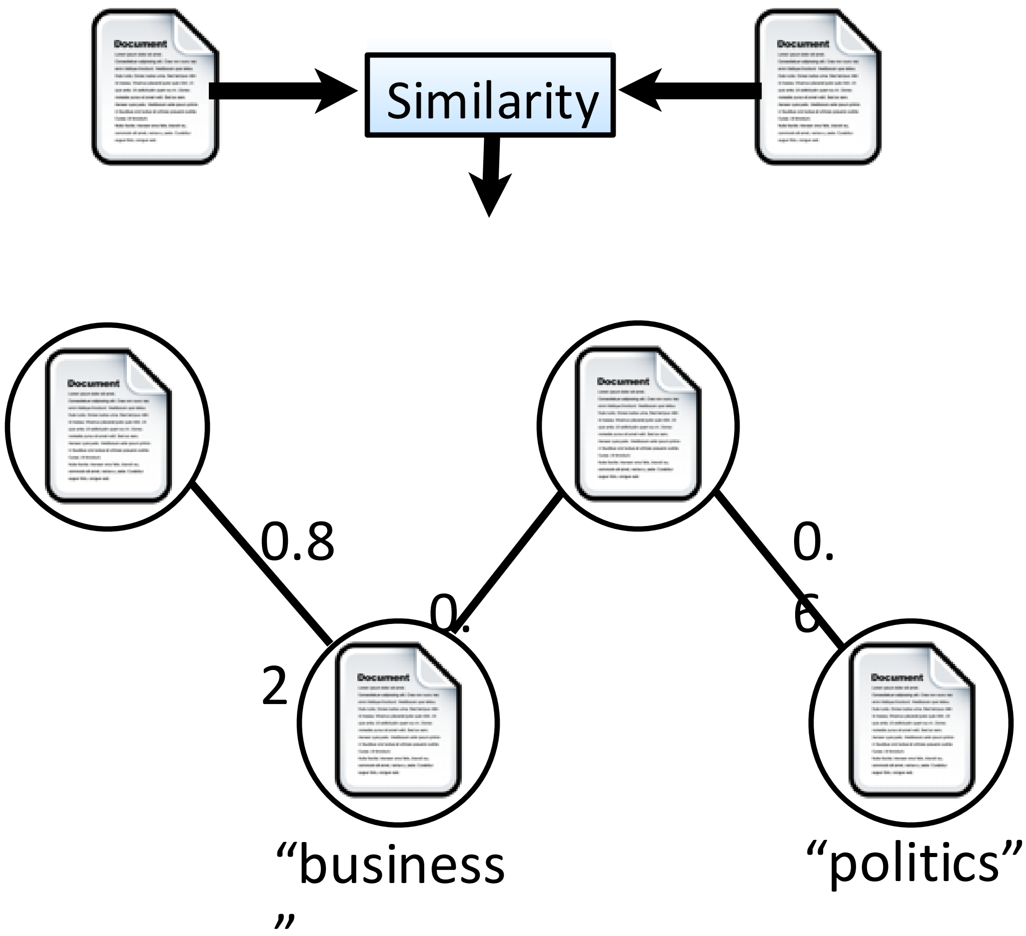
Graph-based SSL



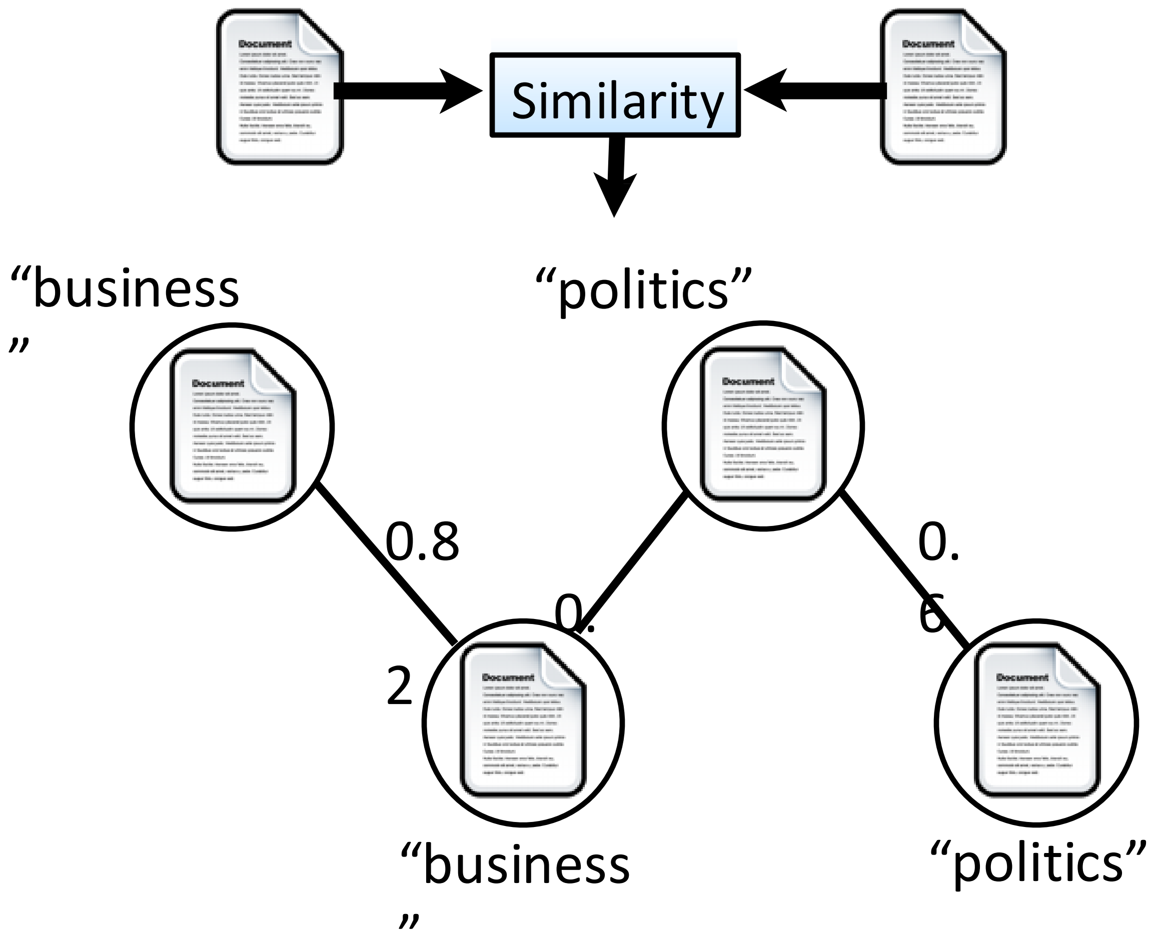
Graph-based SSL



Graph-based SSL



Graph-based SSL

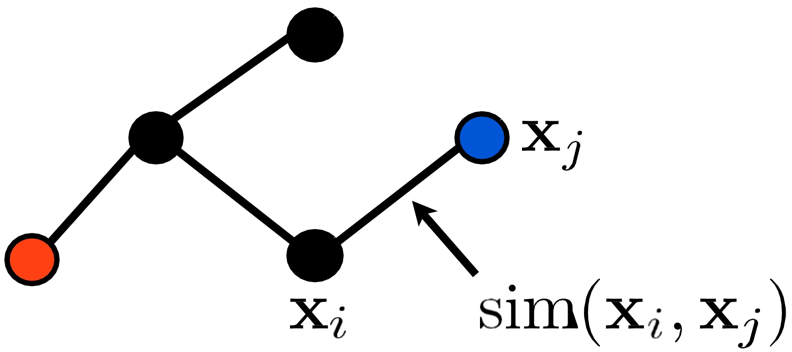


Smoothness/Manifold Assumption

If two instances are similar according to the graph, then output labels should be similar

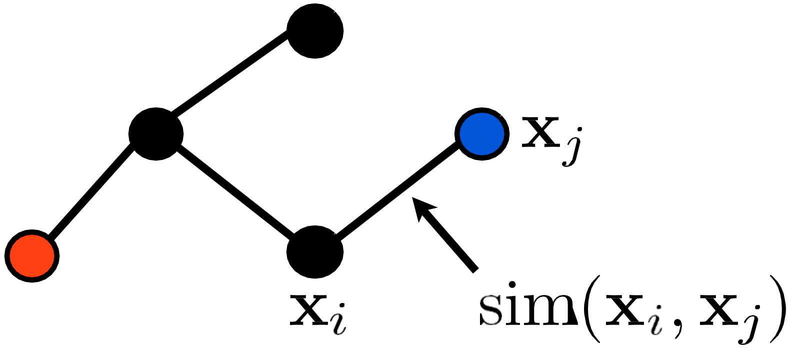
Smoothness/Manifold Assumption

If two instances are <u>similar</u> according to the graph, then <u>output labels</u> should be <u>similar</u>



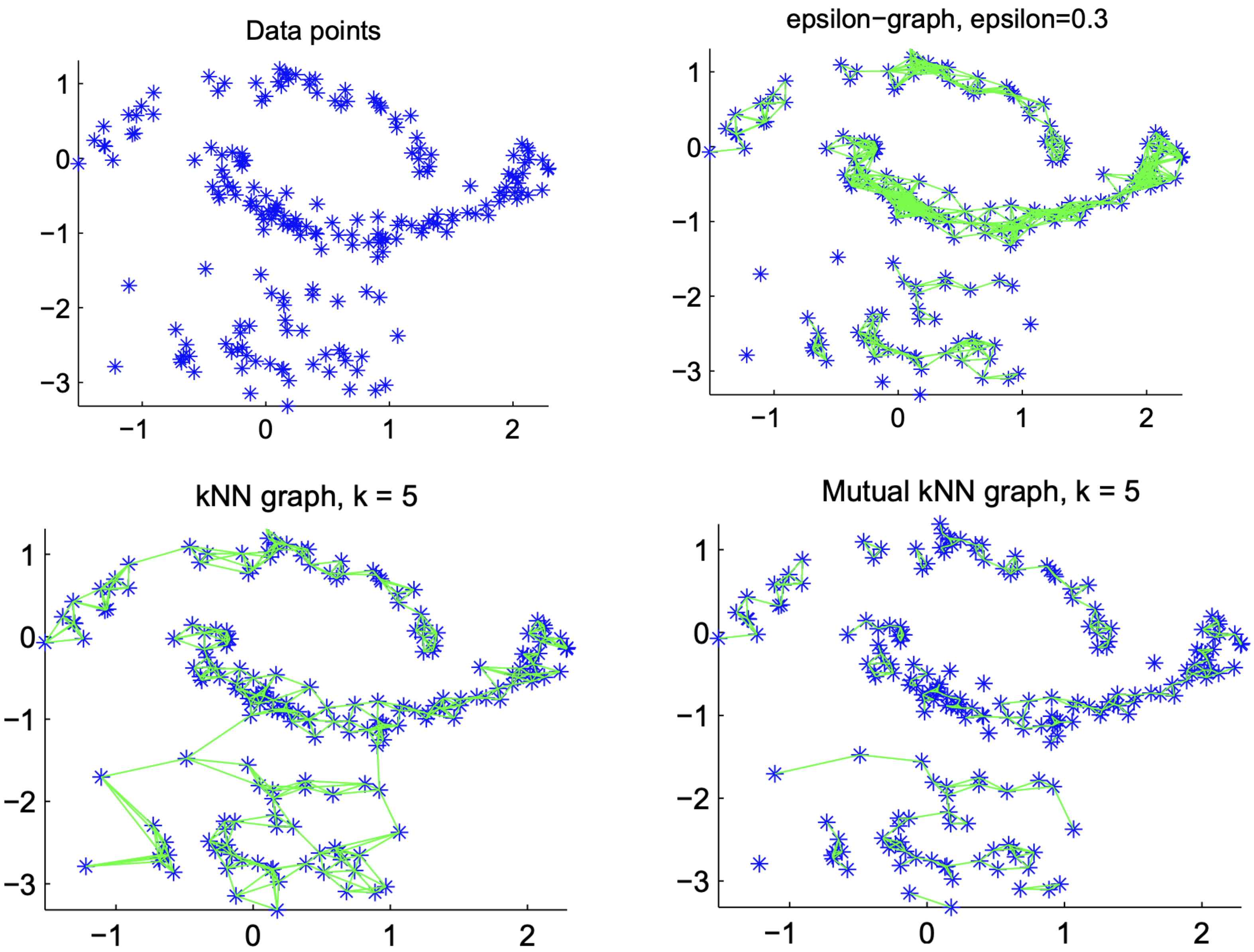
Smoothness/Manifold Assumption

If two instances are similar according to the graph, then output labels should be similar



- Two stages
  - Graph construction (if not already present)
  - Label Inference

Graph Construction



Label Inference Methods

- Label Propagation
- Belief Propagation
- Manifold Regularization
- Spectral Graph Transduction
- Graph Neural Networks

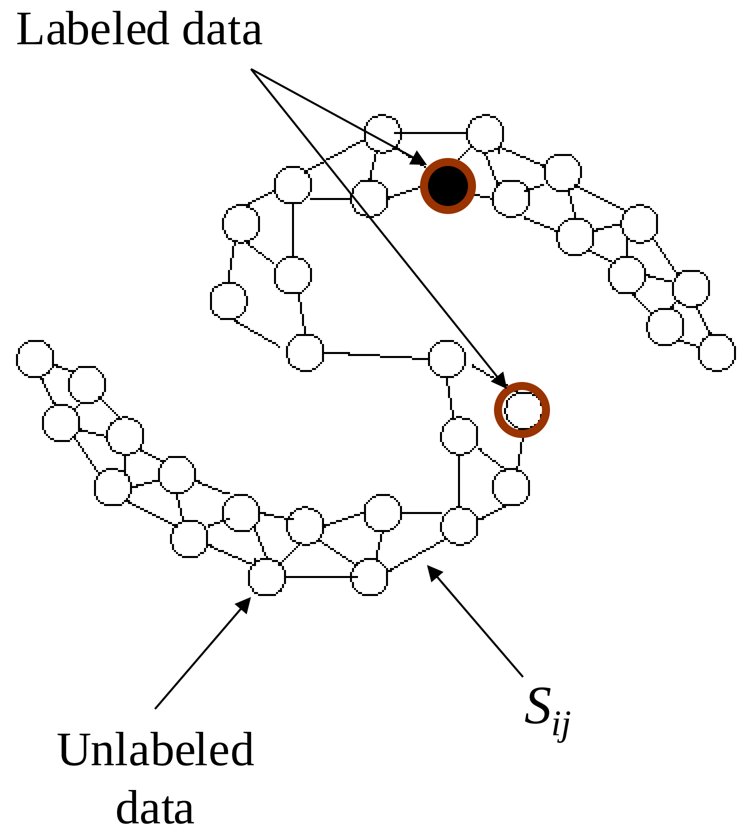
Label Inference Methods

- Label Propagation
- Belief Propagation
- Manifold Regularization
- Spectral Graph Transduction
- Graph Neural Networks

Label Propagation:
Label Spreading

Label Spreading

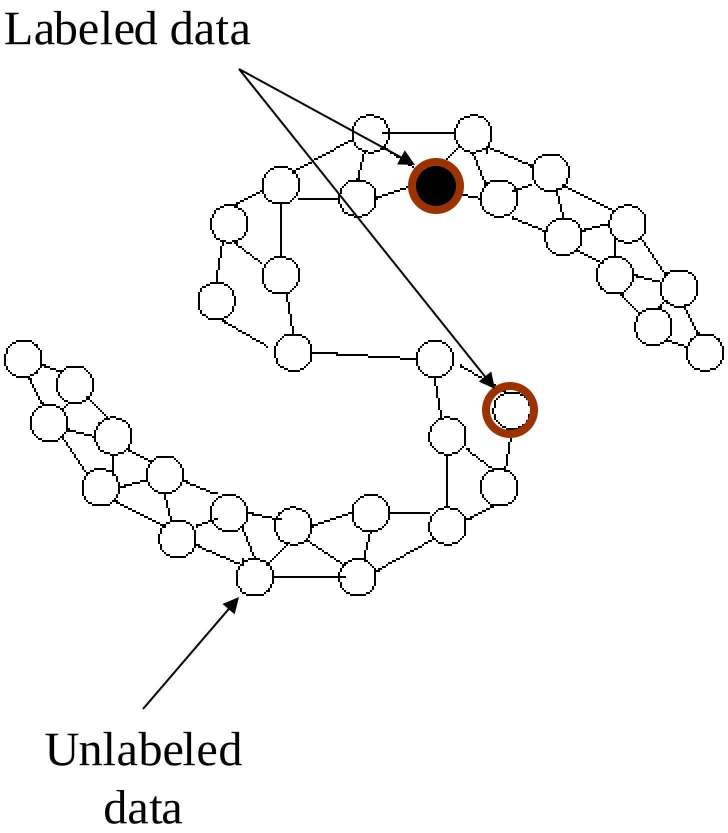
- Each node in the similarity graph is a data point
- Compute the pairwise similarity S_{ij} between data points i and j
- How to predicate labels for unlabeled nodes in the graph?



Label Spreading

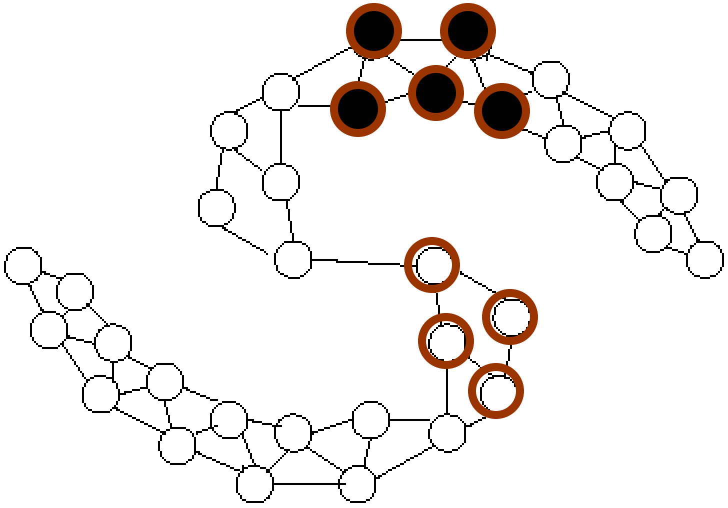
- Idea: Iteratively propagate the labels of the labeled nodes among the graph to **their neighbors** until convergence

- Classification: final label status to predict labels of unlabeled nodes



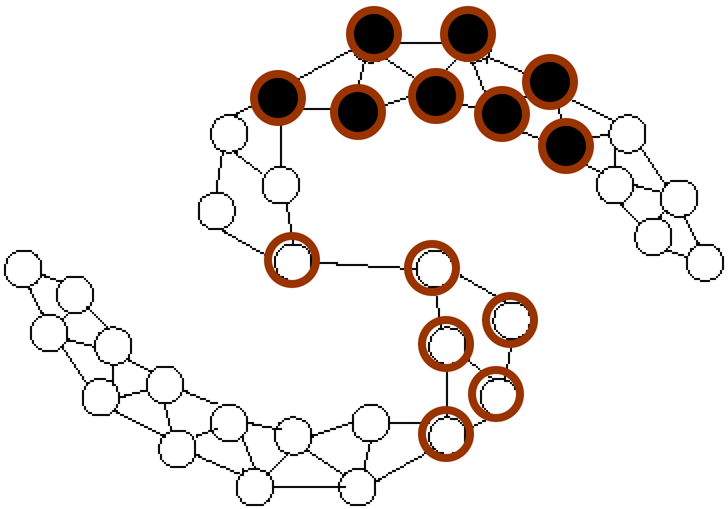
Label Spreading

- First propagation



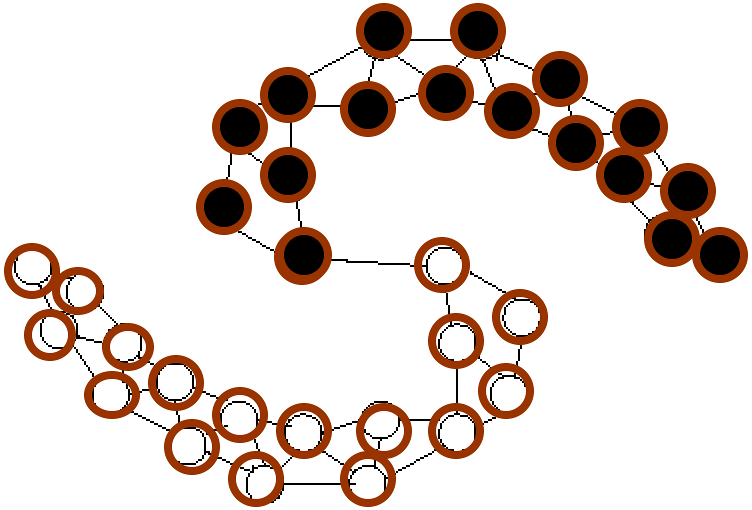
Label Spreading

- Second propagation



Label Spreading

- Convergence



Label Spreading

- Let **S** be the similarity matrix $\mathbf{S}=[\mathbf{S}_{i,j}]_{nxn}$
- Let **D** be a diagonal matrix where $\mathbf{D}_i = \sum_{i \neq j} \mathbf{S}_{i,j}$
- Compute normalized similarity matrix $\mathbf{S'}=\mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$

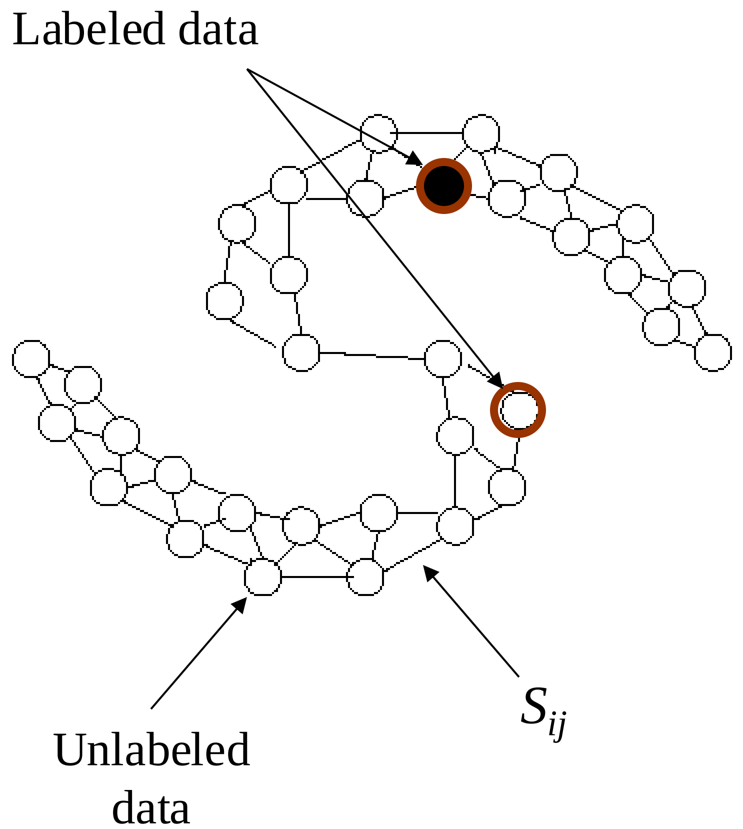
- Let Y be the initial assignment of node labels
  - $Y_i = 1$ when the i-th node is assigned to the ***positive*** class
  - $Y_i = -1$ when the i-th node is assigned to the ***negative*** class
  - $Y_i = 0$ when the i-th node is unlabeled

- Let F be the predicted node labels
  - The i-th node is assigned to the ***positive*** class if $F_i > 0$
  - The i-th node is assigned to the ***negative*** class if $F_i < 0$

Label Spreading

- Initialization

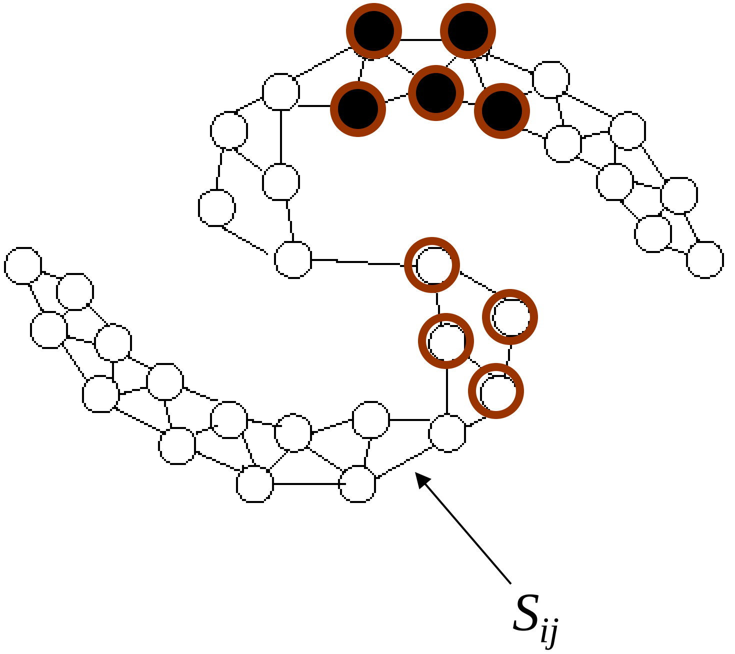
$$F(0) = Y$$



Label Spreading

- First propagation

$F(1) = \qquad \mathbf{S}F(0)$



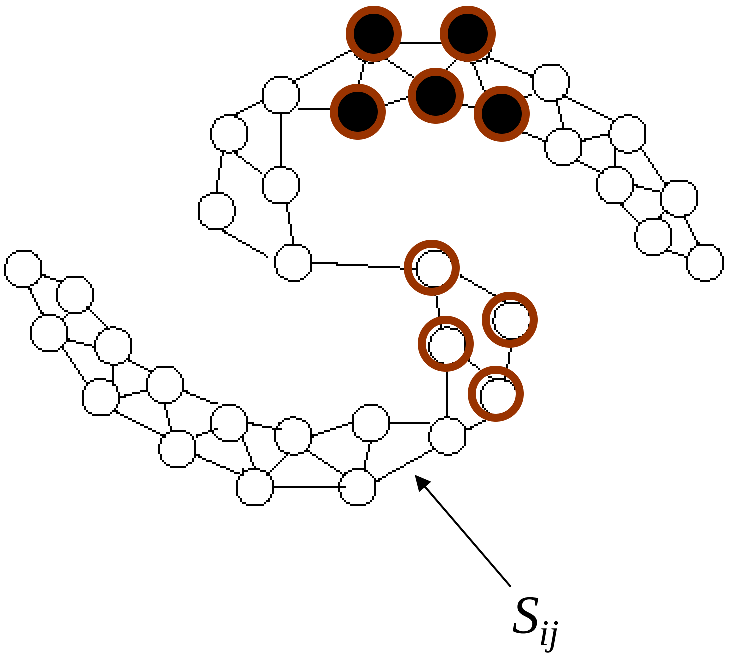
Label Spreading

- First propagation

$$F(1) = (1-\alpha)Y + \alpha\mathbf{S}F(0) \qquad {\color{red}0<\alpha<1}$$

$$= (1-\alpha)Y + \alpha\mathbf{S}Y$$

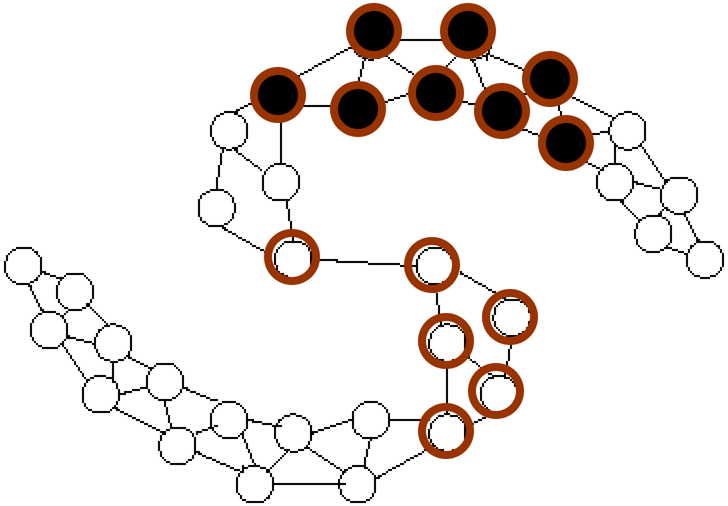
Decay parameter



Label Spreading

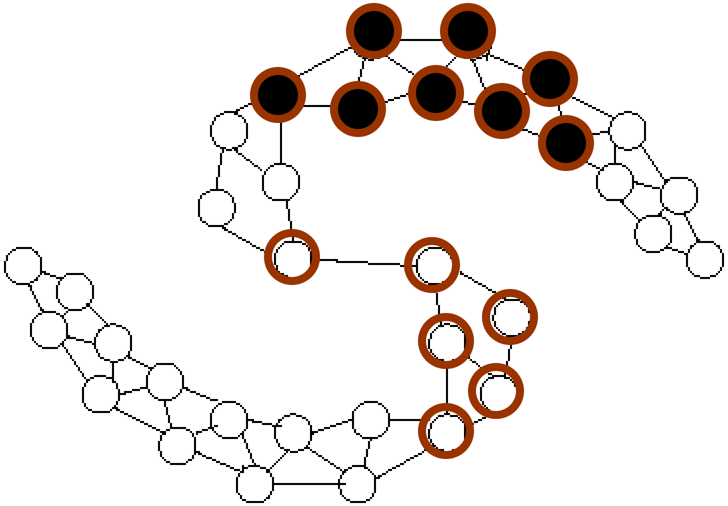
- Second propagation

$F(2) = ?$



Label Spreading

- Second propagation



$$F(2) = (1-\alpha)Y + \alpha \mathbf{S}F(1)$$

$$= (1-\alpha)Y + \alpha \mathbf{S}((1-\alpha)Y + \alpha \mathbf{S}Y)$$

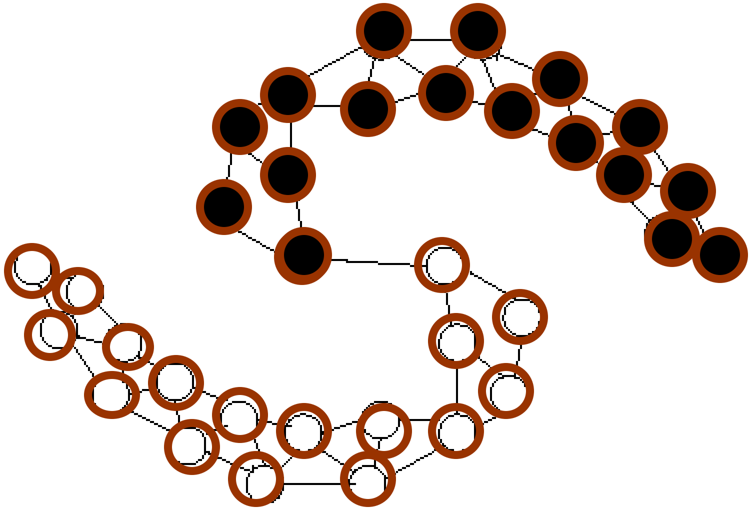
$$= (1-\alpha)Y + \alpha(1-\alpha)\,\mathbf{S}Y + (\alpha \mathbf{S})^2 Y$$

Label Spreading

- t-th propagation

$$F(t) = (1\text{-}\alpha)Y + \alpha\mathbf{S}F(t\text{-}1)$$

$$= (1\text{-}\alpha)\sum_{i=0}^{t-1}(\alpha\mathbf{S})^{i}\,Y + (\alpha\mathbf{S})^{\mathbf{t}}\,Y$$



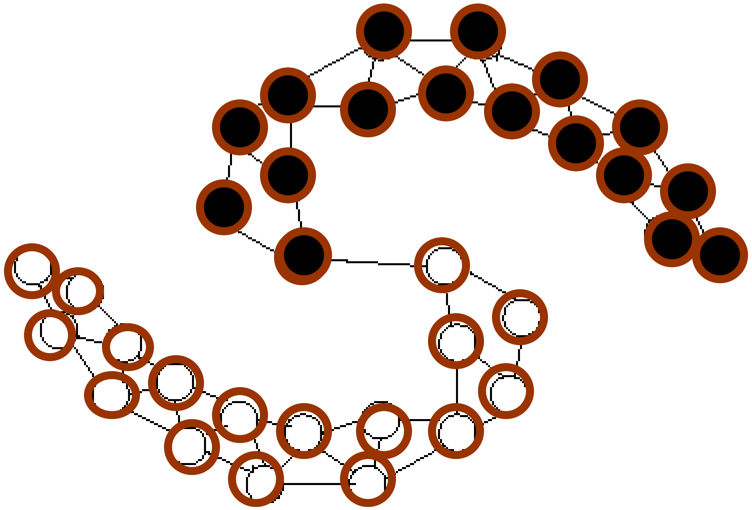
Label Spreading

- Convergence status?

$$\lim_{t\to\infty} \mathbf{F(t)} = ?$$

$$F(t) = (1\text{-}\alpha)Y + \alpha\mathbf{S}F(t\text{-}1) \quad 0<\alpha<1$$

$$= (1\text{-}\alpha)\sum_{i=0}^{t-1}(\alpha\mathbf{S})^i\, Y + (\alpha\mathbf{S})^{\mathbf{t}}\, Y$$



$$\lim_{t\to\infty}(1-\alpha)\sum_{i=0}^{t-1}(\alpha\mathbf{S})^i\, Y = (1-\alpha)(I-\alpha\mathbf{S})^{-1}\, Y$$

$$\lim_{t\to\infty}(\alpha\mathbf{S})^{\mathbf{t}} = 0$$

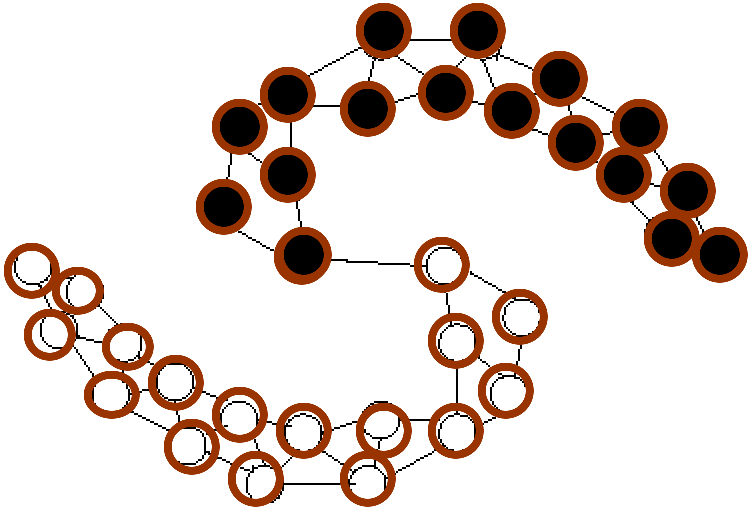
Label Spreading

- Convergence status?

$$\lim_{t\to\infty} \mathbf{F(t)} = ?$$

$$F(t) = (1\text{-}\alpha)Y + \alpha\mathbf{S}F(t\text{-}1) \quad 0<\alpha<1$$

$$= (1\text{-}\alpha)\sum_{i=0}^{t-1}(\alpha\mathbf{S})^i\, Y + (\alpha\mathbf{S})^{\mathbf{t}}\, Y$$

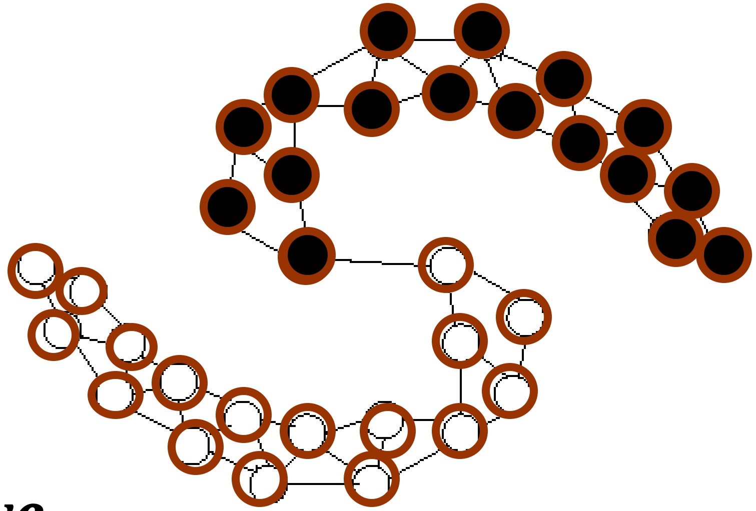


$$\lim_{t\to\infty} \mathbf{F(t)} = (1-\alpha)(I - \alpha\mathbf{S})^{-1}\, Y$$

Label Spreading for Classification

$$\lim_{t\to\infty} \mathbf{F(t)} = (1-\alpha)(I - \alpha\mathbf{S})^{-1}\,Y$$

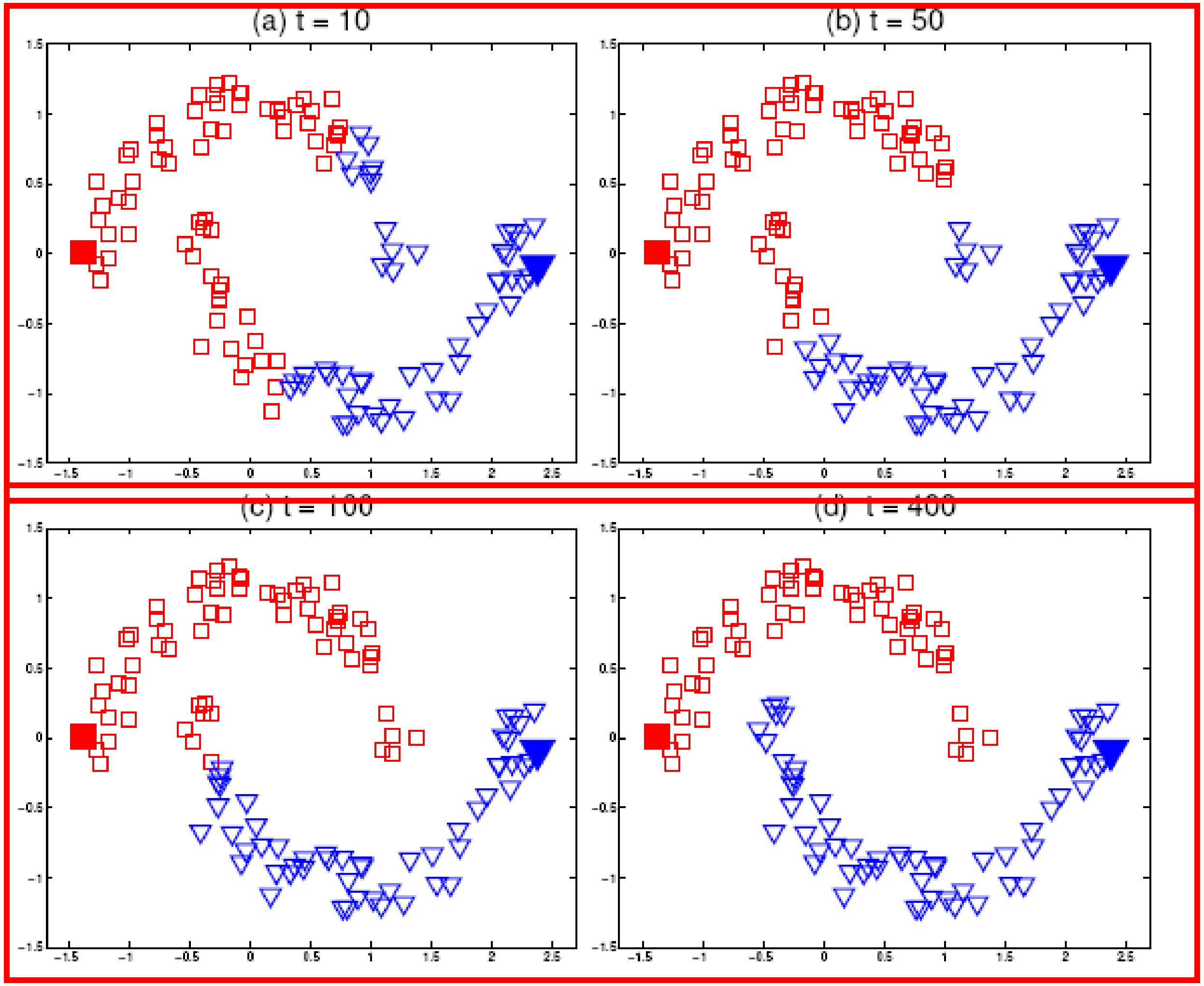
$$\mathbf{F*} = (I - \alpha\mathbf{S})^{-1} Y$$



i-th node is assigned to the ***positive*** ***(negative)*** class if $F^*_i > 0$ (< 0)

Local and Global Consistency

[Zhou et.al., NIPS 03]



Local consistency:

Like KNN

Global consistency:

Beyond KNN

Summary

- Construct a graph using pairwise similarities
- Propagate nodes labels along the graph
- Key parameters
  - α: the decay of propagation
  - S: similarity matrix
- Computational complexity
  - Matrix inverse: O(#all data^3)
  - Cholesky decomposition

Label Propagation: Energy Minimization

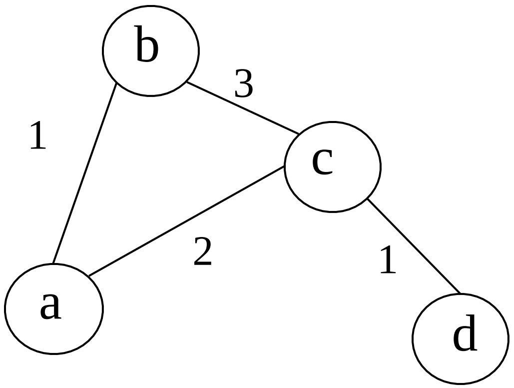
Graph Laplacian

- Laplacian (un-normalized) of a graph:

$$L = D - S, \text{ where } D_{ii} = \sum_j S_{ij} \qquad D_{ij} = 0$$

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & & & & \\ b & & & & \\ c & & & & \\ d & & & & \end{array}$$

$$\begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} \mathbf{3} & \mathbf{-1} & \mathbf{-2} & \mathbf{0} \\ \mathbf{-1} & \mathbf{4} & \mathbf{-3} & \mathbf{0} \\ \mathbf{-2} & \mathbf{-3} & \mathbf{6} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{1} \end{pmatrix} \end{array}$$



L is positive semi-definite

Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

$$\mathrm{f}^{T} L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$

Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

$$\mathrm{f}^{T} L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$

Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

$$\mathrm{f}^{T} L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$

Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

Vector of scores for single label on nodes

$$\mathrm{f}^{T} L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$

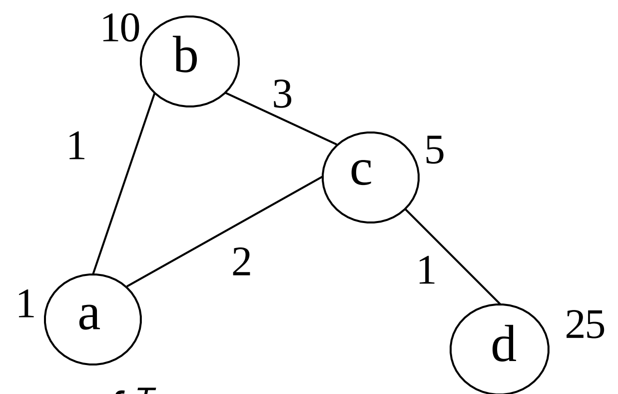
Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

Vector of scores for single label on nodes

$$f^T L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$



$f^T = [1\ 10\ 5\ 25]$

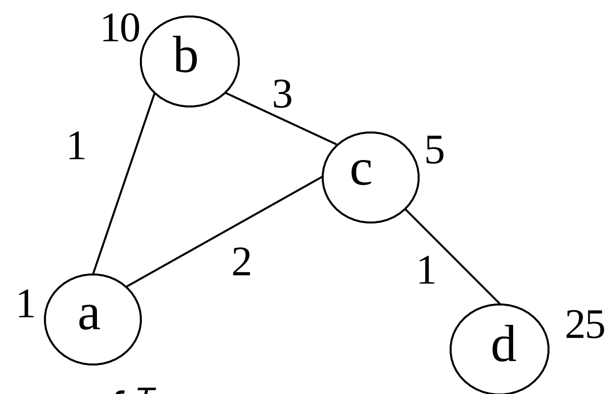
Graph Laplacian (contd.)

- Smoothness of prediction f over the graph in terms of the Laplacian:

Measure of Non-Smoothness

Vector of scores for single label on nodes

$$f^T L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$



$f^T = [1\ 10\ 5\ 25]$

$f^T L f = 588$ Not Smooth

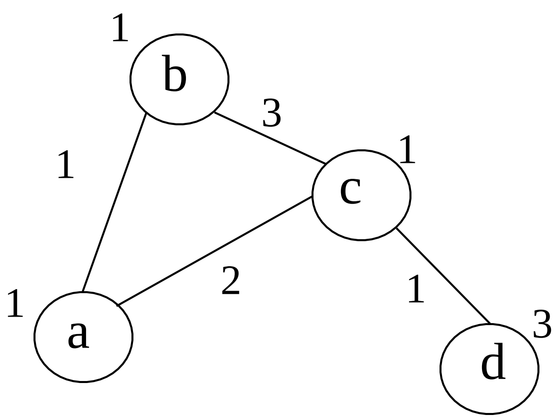
Graph Laplacian (contd.)

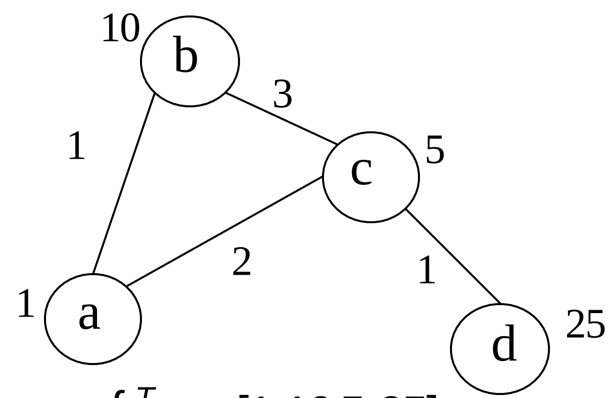
- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

Vector of scores for single label on nodes

$$f^T L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$





f^T = [1 10 5 25]

$f^T L f$ = 588

Not Smooth

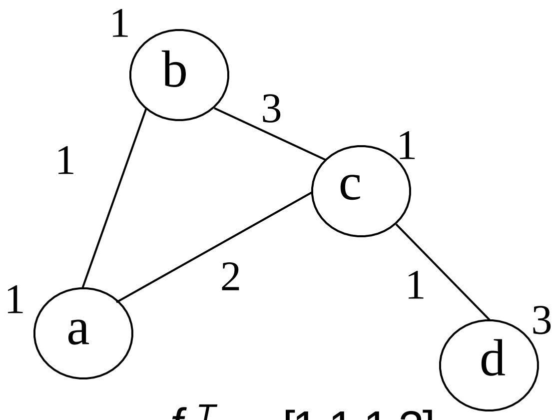
Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Measure of Non-Smoothness

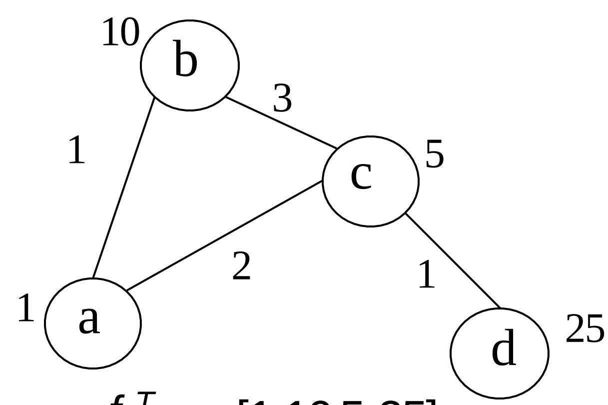
Vector of scores for single label on nodes

$$f^T L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$



$f^T = [1\ 1\ 1\ 3]$

$f^T L f = 4$



$f^T = [1\ 10\ 5\ 25]$

$f^T L f = 588$

Not Smooth

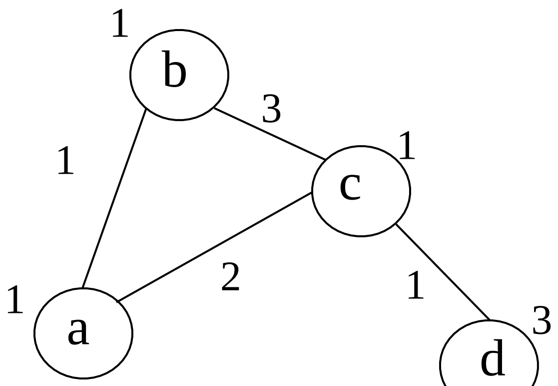
Graph Laplacian (contd.)

- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Vector of scores for single label on nodes

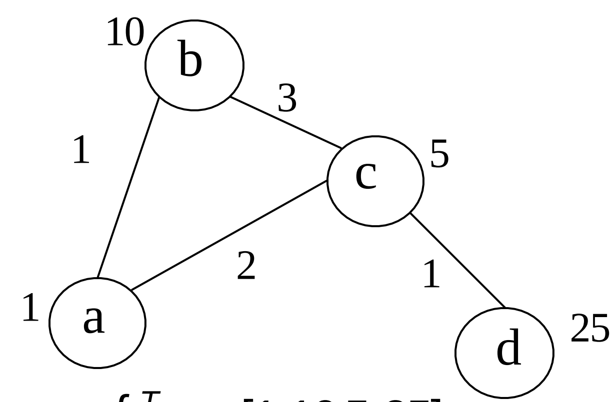
Measure of Non-Smoothness

$$f^{T} L f = \sum_{ij} S_{ij} (f_i - f_j)^2$$



f^T = [1 1 1 3]

$f^T L f$ = 4 Smooth



f^T = [1 10 5 25]

$f^T L f$ = 588 Not Smooth

Relationship between Eigenvalues of the Laplacian and Smoothness

$$Lf = \lambda f$$

$$f^T Lf = \lambda f^T f$$

$$f^T Lf = \lambda$$

Relationship between Eigenvalues of the Laplacian and Smoothness

Eigenvector of L

Eigenvalue of L

$$Lf = \lambda f$$

$$f^T L f = \lambda f^T f$$

$$f^T L f = \lambda$$

Relationship between Eigenvalues of the Laplacian and Smoothness

Eigenvector of L

Eigenvalue of L

$$Lf = \lambda f$$

$$f^T L f = \lambda \boxed{f^T f}$$

= 1, as eigenvectors are are orthonormal

$$f^T L f = \lambda$$

Relationship between Eigenvalues of the Laplacian and Smoothness

Eigenvector of L

Eigenvalue of L

$$Lf = \lambda f$$

$$f^T L f = \lambda \boxed{f^T f}$$

= 1, as eigenvectors are are orthonormal

$$f^T L f = \lambda$$

Measure of Non-Smoothness (previous slide)

Relationship between Eigenvalues of the Laplacian and Smoothness

Eigenvector of L

Eigenvalue of L

$$Lf = \lambda f$$

$$f^T Lf = \lambda \boxed{f^T f}$$

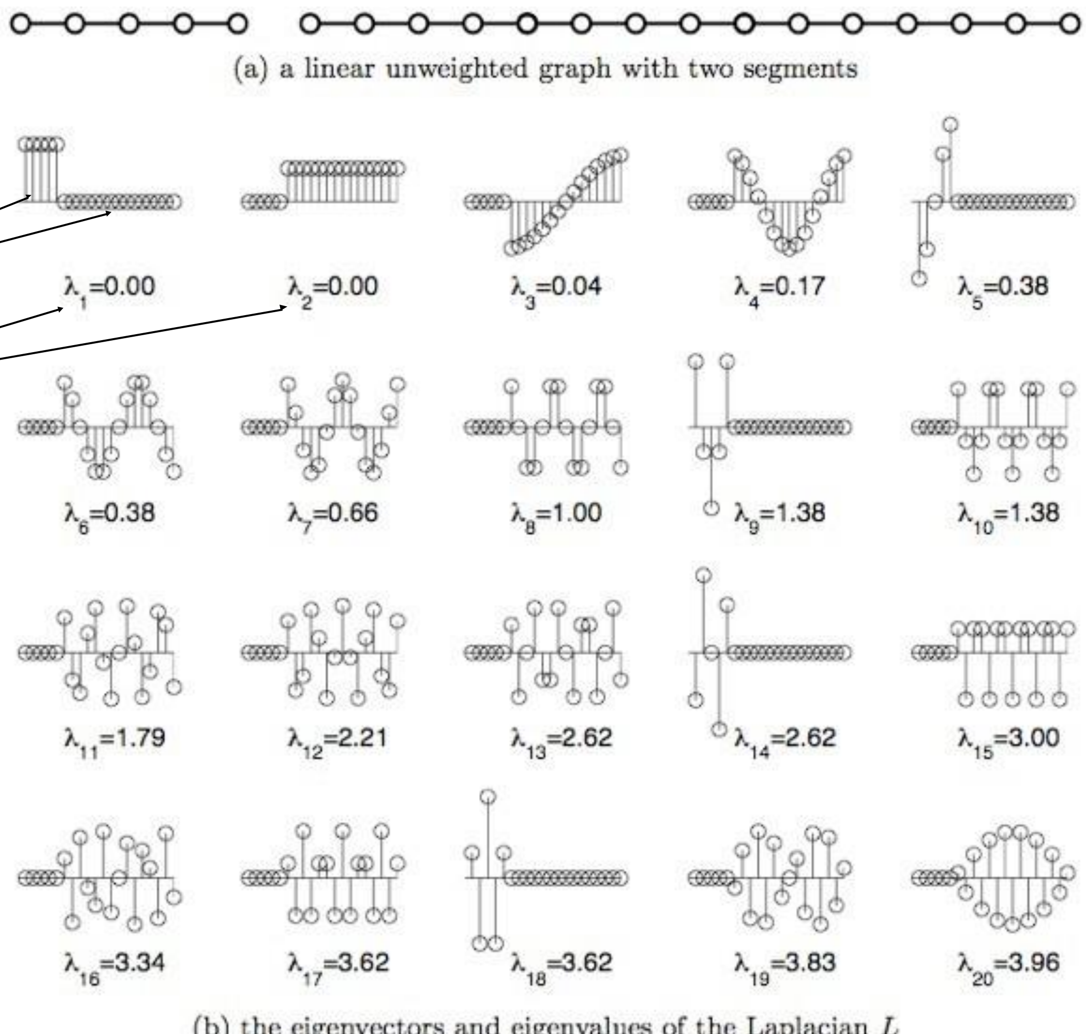
= 1, as eigenvectors are are orthonormal

$$f^T Lf = \lambda$$

Measure of Non-Smoothness (previous slide)

If an eigenvector is used to classify nodes, then the corresponding eigenvalue gives the measure of non-smoothness

Spectrum of the Graph Laplacian



Constant within component

Number of connected components = Number of 0 eigenvalues

Higher Eigenvalue, Irregular Eigenvector, Less smoothness

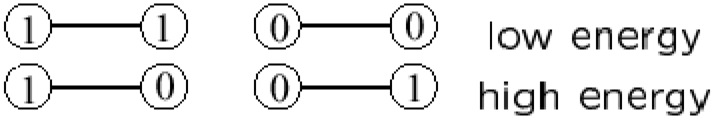
Figure from [Zhu et al., 2005]

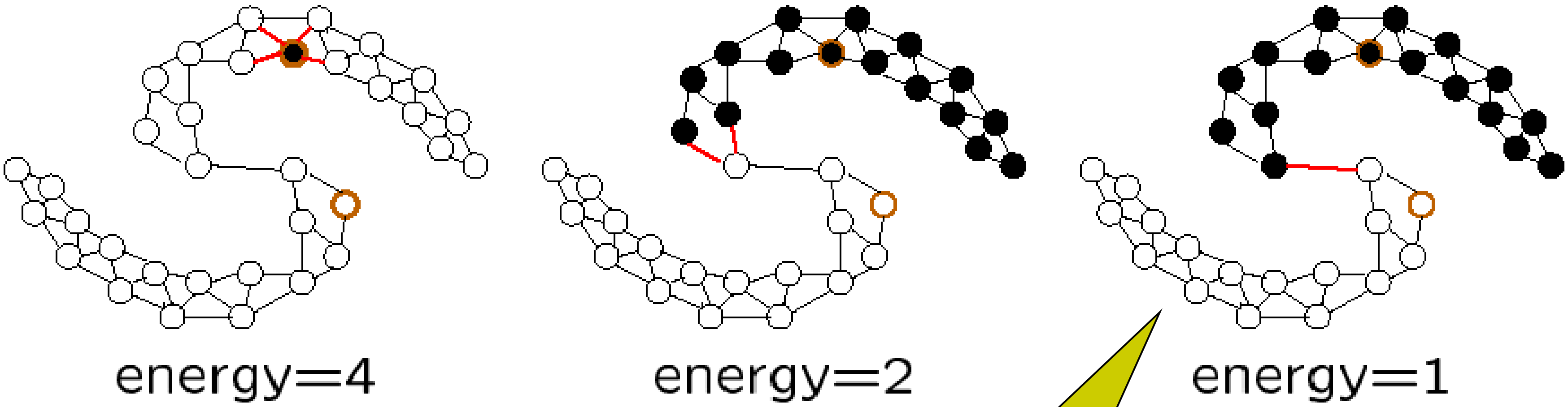
Energy Minimization

- Achieving smoothness ⇔ Minimizing energy
- Energy: $E(F) = \sum_{i,j} S_{i,j}(F_i - F_j)^2$
- Goal: find label assignment F that is
  - minimizes the energy function $E(F)$
  - consistent with labeled examples Y

Low Energy Implies Label Propagation

$E(F) = \sum_{i,j} S_{i,j}(F_i - F_j)^2$





Final classification results

Solution: Harmonic Function

- Min $E(F) = \sum_{i,j} S_{i,j} (F_i - F_j)^2 = F^T(\mathbf{D}\text{-}\mathbf{S})F = F^T\mathbf{L}F$
- Graph Laplacian $\mathbf{D} - \mathbf{S} = \mathbf{L} = \begin{pmatrix} \mathbf{L}_{ll} & \mathbf{L}_{ul} \\ \mathbf{L}_{lu} & \mathbf{L}_{uu} \end{pmatrix}$
- Minimizer for E(F) should be

$$\mathbf{LF} = \mathbf{0}$$

Harmonic function

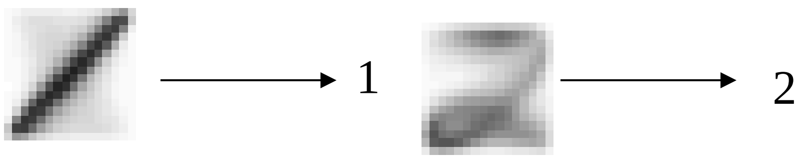
Solution: Harmonic Function

- F should be also consistent with labeled nodes in Y
- Let $F^T = (F_l^T, F_u^T)$, $Y^T = (Y_l^T, Y_u^T)$
- **$F_l = Y_l$**

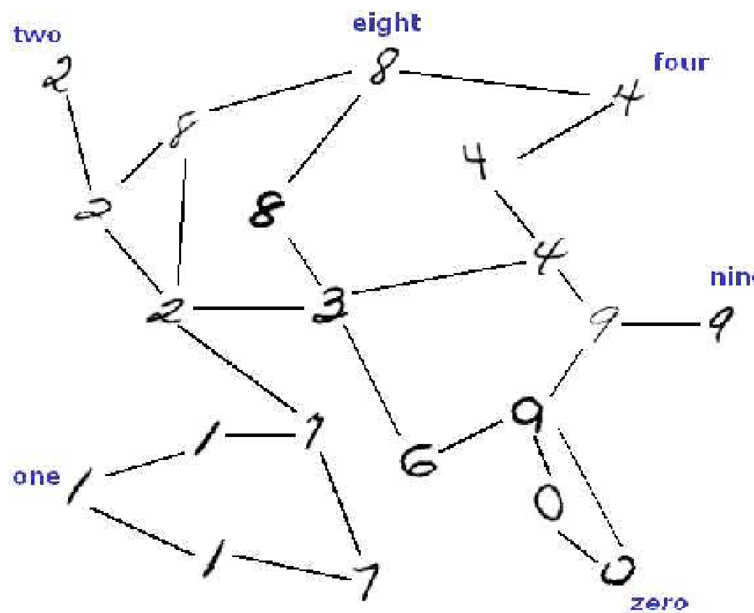
$$\mathbf{L}F = \begin{pmatrix} \mathbf{L}_{ll} & \mathbf{L}_{ul} \\ \mathbf{L}_{lu} & \mathbf{L}_{uu} \end{pmatrix} \begin{pmatrix} Y_l \\ F_u \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{ll} Y_l + \mathbf{L}_{ul} F_u \\ \boxed{\mathbf{L}_{ul} Y_l + \mathbf{L}_{uu} F_u} \end{pmatrix} = 0 \longrightarrow F_u = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} Y_l$$

Optical Character Recognition

- Given an image of a digit letter, determine its value



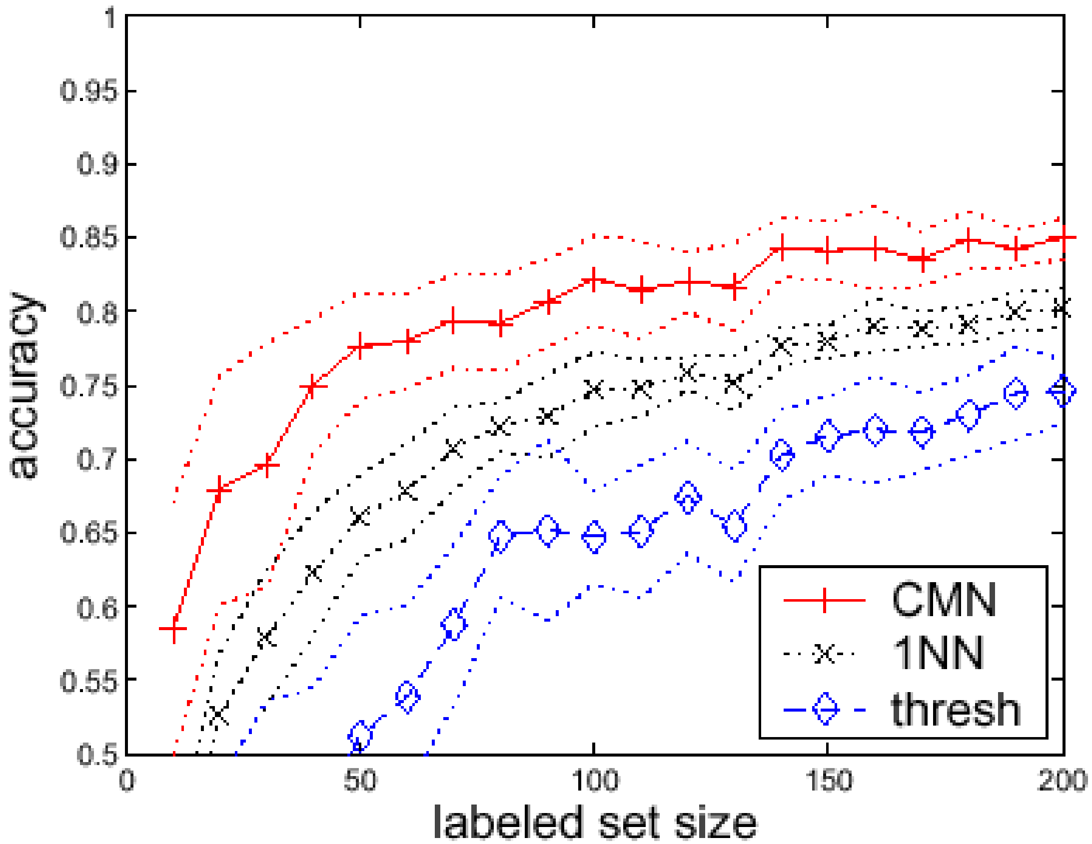
- Create a graph for images of digit letters



Optical Character Recognition

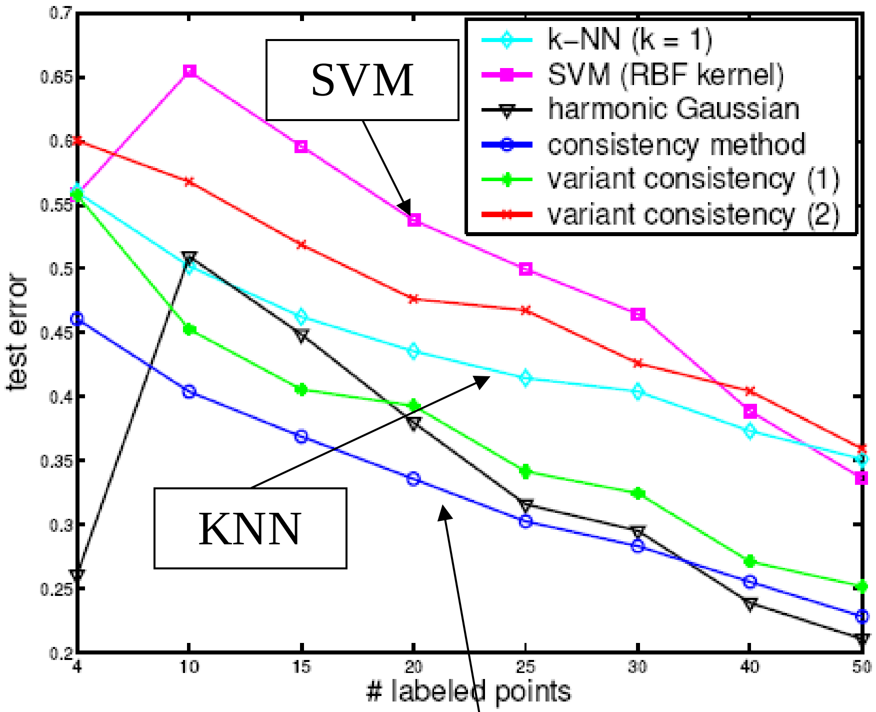
- #Labeled_Examples+#Unlabeled_Examples = 4000

- CMN: label propagation
- 1NN: for each unlabeled example, using the label of its closest neighbor



Application: Text Classification

[Zhou et.al., NIPS 03]



- 20-newsgroups
  - *autos*, *motorcycles*, *baseball*, and *hockey* under *rec*
- Pre-processing
  - stemming, remove stopwords & rare words, and skip header
- #Docs: 3970, #word: 8014

Summary

- Construct a graph using pairwise similarities
- Propagate nodes labels along the graph
  - Energy minimization (achieving smoothness)
- Key parameters
  - S: similarity matrix
- Computational complexity
  - Matrix inverse: O(#unlabeled data^3)
  - Cholesky decomposition